# Accelerating the Computation of Empirical Gramians and Related Methods

Christian Himpe*, Mario Ohlberger*
*University of Münster, Münster, Germany

***Abstract*— Empirical gramians are a tool for nonlinear model reduction. Together with other snapshot-based methods, empirical gramians face several computational issues; among them is the calculation of the snapshots or the feature extraction from the associated gramian matrices. We present an assortment of techniques to accelerate the computation and usage of empirical gramians and demonstrate their advantages.**

## I. Introduction

For linear state-space systems,

$$\dot{x}(t) = Ax(t) + Bu(t),\ \ y(t) = Cx(t),\ \ x(0) = x_0,$$

many model reduction algorithms are established, for example balanced truncation and approximate balancing which partition the state-space based on the input-output energy transfer. For general nonlinear parametrized systems,

$$\dot{x}(t) = f(x(t), u(t), \theta),\ \ y(t) = g(x(t), u(t), \theta),\ \ x(0) = x_0,$$

fewer (computable) model reduction algorithms are available, among them empirical gramian-based model reduction.

## II. Empirical Gramians

Empirical gramians extend the concept of balanced truncation to nonlinear systems [1]. Related snapshot-based methods are also found in fluid dynamics, see for example [2].

All balancing related methods are based upon the controllability operator $\mathcal{C}$ and observability operator $\mathcal{O}$,

$$\mathcal{C}(u) := \int_{-\infty}^{0} e^{-At} B u(t) dt, \quad \mathcal{O}(x_0) := C e^{At} x_0,$$

of which the former maps (past) inputs to states and the latter maps states to (future) outputs.

The system gramians: controllability gramian $W_C$, observability gramian $W_O$ and cross gramian $W_X$, are defined as follows:

$$W_C := \mathcal{C}\mathcal{C}^*, \quad W_O := \mathcal{O}^*\mathcal{O}, \quad W_X := \mathcal{C}\mathcal{O}.$$

Now, the empirical gramians are computed by approximating the operators $\mathcal{C}$ and $\mathcal{O}$ by discrete time-series snapshots. This approach is also valid for nonlinear systems (around a steady state), since the computation requires only (output) trajectory snapshots.

## III. Improved Runge-Kutta Methods

Empirical gramians are computed from trajectories. To obtain trajectories for nonlinear systems a common class of solvers are Runge-Kutta or one-step methods. Explicit Runge-Kutta integrators of order $s$,

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i,\ \ k_i = f(t_n + c_i h, y_n + h(\sum_{j=1}^{i} a_{ij} k_j)).$$

require a repeated evaluation of the system's vector field $f$, which in the nonlinear case, consumes the dominant fraction of computational time. Thus, reducing the number of vector field evaluations expedites the computation of the trajectories. A second class of solvers are multi-step solvers, which reuse evaluations of $f$ from previous steps; for example the explicit Adams-Bashforth methods of order $s$:

$$y_{n+1} = y_n + h \sum_{i=0}^{s} b_i f(t_{n-j}, y_{n-j}).$$

Combining explicit one-step Runge-Kutta methods with explicit multi-step solvers allows higher order solutions with less vector field evaluations.

The accelerated Runge-Kutta methods [3], are two-step Runge-Kutta methods that include information from the previous time step. The improved Runge-Kutta methods [4] are a further development of these two-step methods:

$$y_{n+1} = y_n + h(b_1 k_1 - b_{-1} k_1 + \sum_{i=2}^{s} b_i (k_i - k_{-i})).$$

Like all multi-step methods these two-step Runge-Kutta methods require a starting value, which can be computed by the (minimal local error) Ralston methods [5].

## IV. Generalized Transposition

The empirical gramians are computed as inner products of the centered trajectory snapshots components. In case of the empirical cross gramian by:

$$W_X = \sum_{k=1}^{\dim(u(t))} \int_0^{\infty} \widetilde{W}_X \,\mathrm{d}\, t,\ \ \widetilde{W}_{X\,i,j} = \langle x_i^k(t), y_l^j(t) \rangle.$$

Arranging the $j = 1, \ldots, \dim(x(t))$ (discrete) output trajectory snapshots $y_l^j(t)$ in a 3rd order tensor $\boldsymbol{Y}_{ltj} = y_l^j(t)$, allows a vectorized component-wise dot-product with the discrete state trajectories $x_{it}^k$ by using generalized transpositions:

$$\widetilde{W}_X = x^k \boldsymbol{Y}^{<[231]>}(l, :, :).$$

## V. Approximate Inverse

Gramian-based methods for parameter identification and reduction are available, too; in example the cross-gramian-based approach described in [6]. For a parametrized (square) system augmented with associated "parameter states":

$$\begin{pmatrix}\dot{x}\\ \dot{\theta}\end{pmatrix} = \begin{pmatrix}f(x(t),u(t),\theta)\\ 0\end{pmatrix}, \quad \begin{pmatrix}x(0)\\ \theta(0)\end{pmatrix} = \begin{pmatrix}x_0\\ \theta\end{pmatrix},$$
$$y = g(x(t),u(t),\theta),$$

the empirical cross gramian of this system is called (empirical) joint gramian,

$$W_J := \begin{pmatrix}W_X & W_M\\ 0 & 0\end{pmatrix}.$$

The lower blocks are zero, and thus do not require memory, due to the uncontrollability of the parameter states. The cross-identifiability gramian is then defined as the Schur complement of the symmetric part of the joint gramian:

$$W_{\ddot{I}} := -\frac{1}{2}W_M^T(W_X + W_X^T)^{-1}W_M,$$

which contains the parameter identifiability (observability).

The extraction of the cross-identifiability gramian $W_{\ddot{I}}$ from the joint gramian $W_J$ utilizes the computation of a Schur-complement, which in turn requires a matrix inversion of the symmetric part of the cross gramian $W_X$. A conventional dense matrix inversion is not feasible for large state spaces. The well known identity of the inverse of a matrix as the limit of the Neumann series $A^{-1} = \sum_{k=0}^{\infty}(\mathbb{1} - A)^k$ justifies the computation of an approximate inverse by a truncated Neumann series. A variant of this approach, from [7], where an additive decomposition into diagonal components $D$ and off-diagonal components $E$ by $A = D+E$ is exploited, yields an approximate, yet computationally cheap, inverse:

$$A^{-1} \approx \sum_{k=0}^{K \ll \infty} (-D^{-1}E)^k D^{-1},$$

which for $(K = 2)$ requires only $O(N^2)$ flops.

## VI. Double Re-Orthogonalized Lanczos

Either, balanced truncation and approximate balancing as well as gramian-based parameter identification require a singular value decomposition (SVD) to assemble the (approximate) balancing projection. A truncated SVD (thin SVD) can be computed using a variant of the Lanczos algorithm. The Lanczos algorithm is an iterative partial tri-diagonalization for symmetric eigenvalue problems that requires only vector operations and matrix-vector products. In each iteration a re-orthogonalization is included to ensure orthogonality [8].

To further enhance the quality for large-scale matrix decompositions, an additional orthogonalization as post-processing step can be added for the left (and right) singular vectors.

| | Time [s] |
|---|---|
| $W_{\ddot{I}}$ (Default) | 8303.0 |
| $W_{\ddot{I}}$ (General. Transp.) | 7600.7 |
| $W_{\ddot{I}}$ (Improved RK) | 5540.5 |
| $W_{\ddot{I}}$ (Accelerated) | 4871.1 |
| Matlab TSVD of $W_{\ddot{I}}$ | 13.000 |
| Lanczos TSVD of $W_{\ddot{I}}$ | 1.000 |

TABLE I
Comparison of computational time for the empirical cross-identifiability gramian $W_{\ddot{I}} \in \mathbb{R}^{1024\times1024}$.

## VII. Numerical Comparison

To assess the presented techniques, the computational times for an (SVD of an) empirical cross-identifiability gramian are compared within Matlab using the empirical gramian framework (`emgr`) [9] for a SISO hyperbolic network model,

$$\dot{x}(t) = A\tanh(K(\theta)x(t)) + Bu(t),$$
$$y(t) = Cx(t),$$

of order $\dim(x(t)) = 1024$ and a parametrized diagonal matrix $K = \mathrm{diag}(\theta) \Rightarrow \dim(\theta) = \dim(x(t))$. For details of the benchmark see the source code `http://j.mp/iwmrrf15`. Benchmarked are the default computation (Ralston RK3 solver, component-wise assembly, inverse) against the improved RK3 method and generalized transpositions individually and in combination including the approximate inverse, additionally the Matlab truncated SVD is tested against the Lanczos truncated SVD.

The results in Table I show the timings for the tested improvements which achieve an overall speed-up of $1.7\times$. For large-scale systems with $\dim(x(t)) > 10^5$ these performance gains can reduce the computational complexity of snapshot-based (parametric) nonlinear model order reduction.

## References


[1] S. Lall, J.E. Marsden and S. Glavaski, "Empirical model reduction of controlled nonlinear systems", Proceedings of the 14th IFAC Congress, vol. F, pp. 473–478, 1999.

[2] S. Bagheri, L. Brandt and D.S. Henningson, "Input-output analysis, model reduction and control of the flat-plate boundary layers", Journal of Fluid Mechanics, 620, pp. 263–298, 2009.

[3] F.E. Udwadia and A. Farahani, "Accelerated Runge-Kutta Methods", Discrete Dynamics in Nature and Society, pp. 1–38, 2008.

[4] F. Rabiei, F. Ismail and M. Suleiman, "Improved Runge-Kutta Methods for Solving Ordinary Differential Equations", Sains Malaysiana 42(11): 1679–1687, 2013.

[5] A. Ralston, "Runge-Kutta Methods with Minimum Error Bounds", Mathematics of Computation, 16(80), pp. 431–437, 1962.

[6] C. Himpe and M. Ohlberger, "Cross-Gramian Based Combined State and Parameter Reduction for Large-Scale Control Systems", Mathematical Problems in Engineering, pp. 1–13, 2014.

[7] M. Wu, B. Yin, A. Vosoughi, C. Studer, J.R. Cavallaro and C. Dick, "Approximate Matrix Inversion for High-Throughput Data Detection in the Large-Scale MIMO Uplink", Proceedings of the IEEE International Symposium of Circuits and Systems, pp. 2155–2158, 2013.

[8] J. Chen and Y. Saad, "Lanczos vectors versus singular vectors for effective dimension reduction", IEEE Transactions on Knowledge and Data Engineering, pp. 1091–1103, 2009.

[9] C. Himpe, "Empirical Gramian Framework", `http://gramian.de`, 2015.