CompAMa

www.compama.co.usb.ve

Bull. Comput. Appl. Math., Vol.2, No.1, 2014

ISSN 2244-8659

A Residual Approach for Balanced Truncation Model Reduction (BTMR) of Compartmental Systems

W. La Cruz *†

CompAMa Vol.2, No.1, pp.7-23, 2014 - Accepted May 10, 2014

Abstract

This paper presents a residual approach of the square root balanced truncation algorithm for model order reduction of continuous, linear and time-invariante compartmental systems. Specifically, the new approach uses a residual method to approximate the controllability and observability gramians, whose resolution is an essential step of the square root balanced truncation algorithm, that requires a great computational cost. Numerical experiences are included to highlight the efficacy of the proposed approach.

Keywords: Model reduction, Lyapunov equation, compartmental systems.

1 Introduction

The problem we face is model order reduction of linear and time-invariant compartmental systems defined by the state equations

$$G : \begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t), \end{cases} \tag{1}$$

*Departamento de Electrónica, Computación y Control, Facultad de Ingeniería, Universidad Central de Venezuela, Caracas 1051-DC, Venezuela (william.lacruz@ucv.ve).

†The author is supported by CDCH-UCV project PG-08-8628-2013/1.

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{y} \in \mathbb{R}^p$ are the state, input and output vectors respectively, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$, and $A = (a_{ij})$ is a compartmental matrix, this means that $a_{ii} < 0$, $a_{ij} > 0$ $(i \neq j)$, and $\sum_{i=1}^{n} a_{ij} \leq 0$, for $i, j = 1, \ldots, n$. The linear compartmental systems arise in the study of dynamical systems in which the state of variables quantities are physically meaningful only if they are non-negative or strictly positive. Examples of this systems are mass balance, economic models, population dynamics, among others [1].

The basic problem of model reduction is as follows: Given a nth order system (1), one wishes to find a reduced order system

$$\hat{G} = \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right] : \begin{cases} \dot{\hat{\mathbf{x}}}(t) = \hat{A}\hat{\mathbf{x}}(t) + \hat{B}\mathbf{u}(t) \\ \hat{\mathbf{y}}(t) = \hat{C}\hat{\mathbf{x}}(t) + \hat{D}\mathbf{u}(t), \end{cases} \tag{2}$$

where $\hat{A} \in \mathbb{R}^{r\times r}$, $\hat{B} \in \mathbb{R}^{r\times m}$, $\hat{C} \in \mathbb{R}^{p\times r}$, $\hat{D} = D$, with $r \ll n$ such that the following properties are satisfied:

1. The behavior of the reduced order system $\hat{G}$ is sufficiently close to the original system G. This means, if using the same input $\mathbf{u}(t)$ for both systems, $\hat{\mathbf{y}}(t)$ should be close to $\mathbf{y}(t)$, that is to say, $\|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|$ is small for a certain norm $\|\cdot\|$.

2. System properties, like *stability*, *passivity*, are preserved.

3. The procedure is *computationally efficient*.

Among the model reduction methods are: Modal truncation [2], Padé approximations [3, 4], Reduction / Subtraction-Guyan [5], Krylov subspace methods [6, 7, 8], balanced truncation [9, 10, 11, 12, 13], among others. The balanced truncation method preserves stability [14] and provides a bound on the approximation error [15], i.e. satisfies 1. and 2. above. For small-to-medium scale problems, balanced truncation method can be implemented efficiently. However, for large-scale settings, exact balancing is expensive to implement because it requires dense matrix factorizations and results in a computational complexity of $\mathcal{O}(n^3)$ and a storage requirement of $\mathcal{O}(n^2)$; hence do not satisfy 3. above.

In this work we present a residual approach of balanced truncation for the model order reduction of the system (1), in order to satisfy 3. above. Specifically, we use `ra_lyap` method (Residual Algorithm for the Lyapunov

Equation) introduced in [16] for solving two Lyapunov equations, which solutions are called controllability and observability gramians and are essential in the square root balanced truncation algorithm (SRBT) [11]. The `ra_lyap` method is a residual algorithm of low storage memory that does not requires of large matrix factorizations. Such characteristics may be attractive when the system (1) is sufficiently sparse (as is the case of the compartmental systems). In essence, `ra_lyap` is a variation and extension of iterative schemes for solving nonlinear systems of equations presented and analyzed in [17] and [18].

The SRBT algorithm, in addition to the Lyapunov equations, also require solving a pair of Cholesky factorizations and a SVD of a certain matrix of order n. It is well-known that the computing Cholesky factorization or the SVD requires a significant numerical linear algebra computational cost, but solving a $n \times n$ Lyapunov equation is equivalent to solving a standard linear system with n^2 unknowns, which has a great computational cost and requires a high storage memory. Thus, by the characteristics of `ra_lyap` given above, its use to find the gramians could improve the performance of the SRBT algorithm, in terms of storage memory and CPU time.

The preliminary numerical results show that the new approach performs efficiently for the model order reduction of linear and time-invariant compartmental systems. Moreover, we note that the CPU time required to compute the Cholesky factorization or the SVD is significantly less than the time of resolution of the Lyapunov equations. We cannot guarantee that this fact always happen for any linear and time-invariant system, but we believe that it is true for linear and time-invariant compartmental systems, whose verification is an open problem.

The remainder of this paper is organized as follows. Section 2 shows a basic description of the square root balanced truncation algorithm. Section 3 describes the `ra_lyap` method and presents the residual approach of the square root balanced truncation algorithm. Compartmental systems examples are included in Section 4 to illustrate the performance of the approach presented. In Section 5 some final remarks are included.

Finally, we explain some terminology and fix the notation used throughout the paper. The Euclidean space of the matrices $\mathbb{R}^{n\times n}$ is equipped with the Frobenius inner product and the induced Frobenius norm, given by $\langle Z, W\rangle = trace(Z^T W)$, and $\|Z\|_F = \sqrt{\langle Z, Z\rangle} = \sqrt{trace(Z^T Z)}$, respectively.

2 Square Root Balanced Truncation Algorithm

The goal of this section is to give a basic description of the SRBT algorithm. Consider the notation

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

called the state space representation for G. As mentioned above, the basic problem of model reduction is find a reduced order system (2), with $r \ll n$ such that the behavior of the reduced order system $\hat{G}$ is sufficiently close to the original system G. A method commonly used for find a reduced order system is the balanced truncation. The balanced truncation is based on truncating the state space transformations with a nonsingular application of the form

$$\mathcal{T} : \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \to \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & \hat{D} \end{array}\right] = \left[\begin{array}{c|c} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} & \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \\ \hline \begin{bmatrix} C_1 & C_2 \end{bmatrix} & D \end{array}\right],$$

where $T \in \mathbb{R}^{n \times n}$ is any invertible matrix, and the matrices of the reduced model are defined as: $\hat{A} = A_{11}$, $\hat{B} = B_1$, $\hat{C} = C_1$, and $\hat{D} = D$. The transformation $\mathcal{T}$ balances the controllability gramian W_c, and the observability gramian W_o, which are the solutions of the Lyapunov equations of the form:

$$AW_c + W_c A^T + BB^T = 0, \tag{3}$$

$$A^T W_o + W_o A + C^T C = 0. \tag{4}$$

The most common method to find the transformation $\mathcal{T}$ is the square root method first proposed in [11]. First find the Cholesky factorizations of the gramians:

$$W_c = L_c^T L_c, \qquad W_o = L_o^T L_o,$$

where the matrices L_c and L_o are upper triangular, and always exist since the gramians are positive semi-definite. Next compute a singular value decomposition (SVD):

$$L_c^T L_o = U \Sigma V^T,$$

where U, V are orthogonal matrices and Σ is a diagonal matrix. The following step is to form the matrix T which requires the system to be controllable and observable, that is, W_c and W_o must be positive semi-definites. This is guaranteed when A is a stable matrix, that is, when all eigenvalues of A have negative real part. Therefore, if A is a stable matrix, then W_c and W_o are both positive semi-definite and T is defined as

$$T = L_c U \Sigma^{-1/2}, \qquad T^{-1} = \Sigma^{-1/2} V^T L_o^T.$$

Now, it is necessary to truncate the original system to define an error bound for the balanced truncation. For a system G with Hankel singular values ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$), which are the square roots of the eigenvalues of $W_c W_o$, the approximation error for a balanced truncation reduced order model of order r, $\hat{G}$, satisfies the inequality

$$2(\sigma_{r+1} + \cdots + \sigma_n) \geq \|G - \hat{G}\|_{H_\infty} \geq \sigma_{r+1},$$

where $\|\cdot\|_{H_\infty}$ is called H_∞-norm. This way, balanced truncation performs a truncating of the states corresponding to the $n-r$ smallest Hankel singular values from a balanced realization (see [19] and [20] for details) and compute the reduced order system (2). Algorithm 2.1 shows the square root balanced truncation algorithm.

Algorithm 2.1 (Square root balanced truncation algorithm)

1: Solve the Lyapunov equations (3) and (4).
2: Compute Cholesky factorizations: $W_c = L_c L_c^T$, $W_o = L_o L_o^T$.
3: Compute SVD: $L_c^T L_o = U \Sigma V^T$.
4: Form the balancing transformation: $T = L_c U \Sigma^{-1/2}$, $T^{-1} = \Sigma^{-1/2} V^T L_o^T$.
5: Form the balanced realization transformations as: $\overline{A} = T^{-1} A T$, $\overline{B} = T^{-1} B$, and $\overline{C} = CT$.
6: Select the reduced model order r so that

$$\overline{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \quad \overline{D} = D,$$

where $A_{11} \in \mathbb{R}^{r \times r}$, $B_1 \in \mathbb{R}^{r \times m}$, $C_1 \in \mathbb{R}^{p \times r}$.
7: Truncate the transformed system to obtain the reduced system $\hat{G}$ with $\hat{A} = A_{11}$, $\hat{B} = B_1$, $\hat{C} = C_1$, and $\hat{D} = D$.

Remark 2.1. *It is noteworthy that the SRBT algorithm in [3], among other balanced truncation methods based on Lyapunov equations, uses the Bartels-Stewart method [21] for solving such equations. The Bartels-Stewart method* *is a direct method that uses the Schur factorization of the matrix A to transform the equation $AX + XA^T = Q$ into a triangular Lyapunov equation (see [21] for details).*

3 Residual Approach for Balanced Truncation

In this section we present the residual approach of the square root balanced truncation algorithm, for model order reduction of continuous, linear and time-invariante compartmental systems. We begin by describing `ra_lyap` method [16] for solving Lyapunov matricial equations of the form

$$MX + XM^T + Q = 0, \tag{5}$$

where $M \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{n\times n}$, and X is the unknown matrix in $\mathbb{R}^{n\times n}$.

In order to describe the `ra_lyap` method, we first need to write (5) as a map $F : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, namely

$$F(X) = MX + XM^T + Q. \tag{6}$$

The main idea of `ra_lyap` method is to generate iterates, from an initial point $X_0 \in \mathbb{R}^{n\times n}$ given, using the residual $F(X)$ as search direction and computing a step-lenght to guarantee the convergence. In other words, from $X_0 \in \mathbb{R}^{n\times n}$ we obtain the iterate X_{k+1} as

$$X_{k+1} = X_k - \lambda_k \alpha_k F(X_k), \quad \text{for } k \geq 0,$$

where $\lambda_k > 0$ is the step-length, $-\alpha_k F(X_k)$ is the search direction, and α_k is the well-known spectral coefficient which is closely related to the Barzilai-Borwein choice of step-length [22] and it is generally defined as

$$\alpha_k = \frac{\langle S_{k-1}, S_{k-1}\rangle}{\langle S_{k-1}, Y_{k-1}\rangle},$$

where $S_{k-1} = X_k - X_{k-1}$, and $Y_{k-1} = F(X_{k-1}) - F(X_k)$. The detailed description of `ra_lyap` method is shown formally in Algorithm 3.1.

The following proposition shows some properties of the sequences $\{X_k\}$ and $\{\alpha_k\}$ generated by `ra_lyap` method. Proposition 3.1 correspond to Proposition 3.2 from [16].

Algorithm 3.1 (`ra_lyap` method)

Require: $X_0 \in \mathbb{R}^{n \times n}$, $\gamma, \sigma, \rho \in (0,1)$, $0 < \alpha_{max} < \infty$, $0 < \alpha_0 < \alpha_{max}$, an integer $k_0 \geq 0$.

$F_0 = MX_0 + X_0M^T + Q$;
$k \leftarrow 0$;
while $\|F_k\|_F > 0$ **do**
  Define $S_k = -\alpha_k F_k$;
  $G_k = MS_k + S_kM^T$;
  Set $\lambda \leftarrow 1$;
  if $k > k_0$ **then**
    while $\|F_k + \lambda G_k\|_F^2 > \|F_k\|_F^2 + \rho^{k-k_0} - \gamma\lambda^2\alpha_k^2\|F_k\|_F^2$ **do**
      Set $\lambda \leftarrow \sigma\lambda$;
    end while
  end if
  Define $\lambda_k = \lambda$, $X_{k+1} = X_k + \lambda_k S_k$, $F_{k+1} = F_k + \lambda_k G_k$;
  Define $\alpha_{k+1} = \min\left(\alpha_{max}\,,\ \alpha_k^2\|F_k\|_F^2/\langle S_k, G_k\rangle\right)$;
  Set $k \leftarrow k + 1$;
end while

Proposition 3.1. *Assume that the symmetric part of M is positive (negative) definite. Let $\{X_k\}$ and $\{\alpha_k\}$ be the sequences generated by* `ra_lyap` *method. Then the following hold, for $k \geq 0$:*

$$\begin{aligned} X_{k+1} &= X_k + \lambda_k S_k, \\ F_{k+1} &= F_k + \lambda_k G_k, \\ \alpha_{k+1} &= \min\left(\alpha_{max}\,,\ \alpha_k^2\|F_k\|_F^2/\langle S_k, G_k\rangle\right), \end{aligned}$$

where $F_k = F(X_k)$, $S_k = -\alpha_k F_k$, $G_k = MS_k + S_kM^T$, and the mapping $F(X)$ is given by (6)*.*

In order to guarantee global convergence, from any initial guess, the step-length $\lambda_k \in (0,1]$ is chosen such that it satisfies the following inequality, called *linear search*,

$$\|F_{k+1}\|_F^2 \leq \|F_k\|_F^2 + \rho^{k-k_0} - \gamma\lambda_k^2\alpha_k^2\|F_k\|_F^2.$$

This globalization strategy is similar to the proposition presented in [18] with some variations. As in [18], the linear search also requires some given parameters: $\gamma, \sigma, \rho \in (0,1)$, $0 < \alpha_{max} < \infty$, $0 < \alpha_0 < \alpha_{max}$, and an integer $k_0 \geq 0$.

Theorem 3.1 correspond to Theorem 3.1 from [16] and it establishes the convergence of Algorithm 3.1.

Theorem 3.1. *Let F be the mapping defined by* (6)*. Assume that the symmetric part of M is positive (negative) definite. Let $\{X_k\}$ be the sequence generated by Algorithm 3.1. Then,* $\lim_{k\to\infty} \|F(X_k)\|_F = 0$.

Let us now describe the residual approach of the square root balanced truncation algorithm. First, let us prove that the `ra_lyap` method applied to the Lyapunov equations (3) and (4) converges to solution of such equations. The following theorem establishes this result. Theorem 3.2 says that the sequences generated by the `ra_lyap` method when is applied to (3) and (4) converges to the controllability and observability gramians, respectively.

Before presenting the proof of Theorem 3.2 is necessary to consider the following elementary result on Lyapunov matrix equations.

Proposition 3.2. *Let M be a real n by n matrix and let $\delta(M)$ be the set of eigenvalues of M. Then the following statements are true:*

i) The Lyapunov equation (5) *has an unique solution for every choice of Q if and only if $\nu + \mu \neq 0$, for all $\nu, \mu \in \delta(M)$.*

ii) If M is stable, and Q is symmetric positive (semi)definite, then the solution of the Lyapunov equation (5) *is symmetric positive (semi)definite.*

Theorem 3.2. *Assume that the system* (1) *is compartmental. Let $\{X_k\}$ be the sequence generated by* `ra_lyap` *method. Then the following hold:*

i) If `ra_lyap` *is applied to* (5) *with $M = A$ and $Q = BB^T$, then the sequence $\{X_k\}$ converges to the controllability gramian W_c.*

ii) If `ra_lyap` *is applied to* (5) *with $M = A^T$ and $Q = C^TC$, then the sequence $\{X_k\}$ converges to the observability gramian W_o.*

Proof. We begin by proving that A is a stable matrix, i.e., all eigenvalues of A have negative real part. Since A is a compartmental matrix, then $a_{ii} < 0$, $a_{ij} > 0$ $(i \neq j)$, and $\sum_{i=1}^{n} a_{ij} \leq 0$, for $i, j = 1, \ldots, n$. Thus, by Gerschgorin's

Theorem, all eigenvalues of A are located in one of the closed discs of the complex plane centered at a_{ii} and having the radius

$$\rho_j = \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} < |a_{jj}|;$$

in other words, for each eigenvalue ω of A there is j such that $|\omega - a_{jj}| < \rho_j < |a_{jj}|$. Therefore, since $a_{jj} < 0$, all eigenvalues of A have negative real part, that is to say, A is a stable matrix.

Proof of i). Since A is a stable matrix, then $\nu + \mu \neq 0$, for all $\nu, \mu \in \delta(A)$. Therefore, for $M = A$, by Proposition 3.2 the equation (5) has an unique solution. Also, since BB^T is positive semi-definite, then the unique solution of (5) is symmetric positive semi-definite for the choice $Q = BB^T$. Thus, for $M = A$ and $Q = BB^T$, from Theorem 3.1 and the continuity of the mapping $F(X)$, we obtain that the sequence $\{X_k\}$ converges to the unique solution of (5). By a similar reasoning we prove *ii)*. □

We are now able to present the residual approach of the square root balanced truncation algorithm. This approach consists on using `ra_lyap` method in Algorithm 3.1 to approximate the controllability and observability gramians. This yields a version of the square root balanced truncated algorithm which will be referred, throughout the rest of this document, as `ra_srbt` method.

4 Numerical Experiments

We now present numerical experiments to show the efficiency of `ra_srbt` method in the model order reduction of linear and time-invariant compartmental systems. In all experiments described here we use Matlab for the implementation of the methods `ra_srbt` and `ra_lyap` (Matlab codes written by the author are available upon request). All the runs were carried out on an Intel Core i5 at 2.4 GHz with 8 GB of RAM.

We take `error_bound = sqrt(eps)` $\approx 1.49 \times 10^{-8}$ as error bound for balanced truncation, that is, the order r of the reduced model satisfies the inequality

$$2(\sigma_{r+1} + \cdots + \sigma_n) \leq \texttt{error\_bound}.$$

We implement Algorithm 3.1 with the following parameters: $k_0 = 100000$, α_{max} = `1/abs(eigs((A+A')/2,1,'sm'))`, $\rho = 0.9999$, $\sigma = 0.5$, $\gamma = 10^{-4}$, and $\alpha_0 = 1$, where the Matlab's code `eigs((A+A')/2,1,'sm')` obtains the eigenvalue of magnitude smaller of the symmetric part of A. In Algorithm 3.1, we take $X_0 = BB^T$ as initial point for solving the equation (3) and $X_0 = C^TC$ for solving the equation (4), and we stopped the process when

$$\|F(X_k)\|_F \leq e_a + e_r\|F(X_0)\|_F,$$

where $e_a = 10^{-14}$ and $e_r = 10^{-15}$. For the simulation, we take $\mathbf{x}(0) = 0$ as initial state.

We compare the performance of `ra_srbt` with the following Matlab's procedure which we have denoted `matlab_bt`.

```
% matlab_bt procedure
sys = ss(A,B,C,D);  % A, B, C and D must be full matrices
[sysb, g] = balreal(sys);  % Compute balanced realization
elim = (g<error_bound);  % Small entries of g are negligible states
sysb = modred(sysb,elim,'Truncate');
```

Here `sys = ss(A,B,C,D)` creates a state-space model object representing the state-space equations (1). The function `[sysb, g] = balreal(sys)` (we use the default values), for stable systems, computes an equivalent realization `sysb` for which the controllability and observability gramians are equal and diagonal, their diagonal entries forming the vector `g` of Hankel singular values. Small entries in `g` indicate states that can be removed to simplify the model. The function `sysb = modred(sysb,elim,'Truncate')` reduces the order of a state-space model `sysb` by eliminating the states found in the vector `elim = (g < error_bound)`, using the state elimination method `Truncated`. The full state vector $\mathbf{x}$ is partitioned as $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2]$, where $\mathbf{x}_2$ is to be discarded and the reduced state is set to $\mathbf{x}_r = \mathbf{x}_1 + T\mathbf{x}_2$. Thus, `matlab_bt` procedure is an implementation of SRBT algorithm. See [10], [9], [21] for details on the balanced realization algorithm used in `balreal`.

Our first example consists on n connected water reservoirs (see [23] for details) as schematically shown in Figure 1, where the compartmental matrix $A = (a_{ij})$ is given by

$$a_{ij} = \frac{\kappa}{a_i}\begin{cases} -d_{o,i}^2 - \sum_{s=1}^n d_{is}^2, & i = j, \\ d_{ij}^2, & i \neq j, \end{cases} \quad \text{with } d_{ii} = 0,$$

$B = (1/a_1\, 0 \cdots 0)^T$, $C = \kappa(d_{o,1}^2 \cdots d_{o,n}^2)$, $D = 0$, the reservoirs R_i and R_j are connected by a pipe of diameter $d_{ij} = d_{ji} \geq 0$, a_i is the base area of the reservoir R_i, the direct flow f_{ij} from R_i to R_j is assumed to be linearly dependent of the pressure difference on both ends, and κ is a constant representing gravity as well as viscosity and density of the medium. In this example $x_i(t)$ represents fill level of the reservoir R_i. We consider the external inflow to reservoir R_1 as the input of the system. The output is given by the sum of all outflows $f_{o,i}$ of R_i through a pipe with diameter $d_{o,i}$.

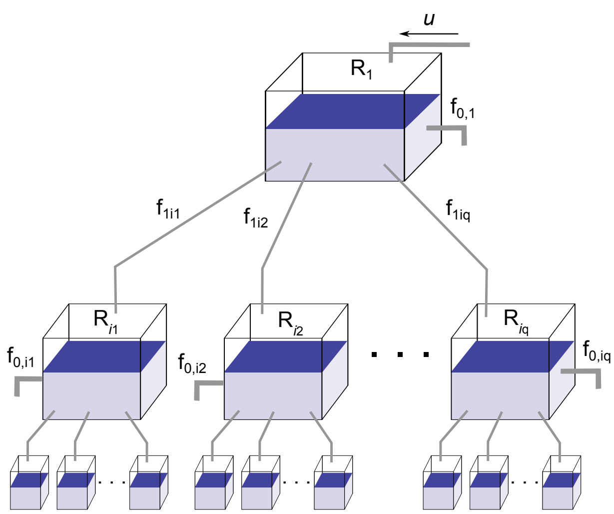

Fig. 1: System of n water reservoirs

For illustration, we have constructed n reservoirs (see Figure 1) in such a way that the reservoir R_1 is connected to q reservoirs, $R_{i_1}, \dots, R_{i_q}$, by a pipe of diameter 1, also those reservoirs are not connected to each other. In turn, each reservoir R_{i_l} is connected to g_{i_l} reservoirs by a pipe of diameter 0.2, and the g_{i_l} reservoirs are not connected to each other. Moreover, for simplicity we set $a_i = 1$ and $\kappa = 1$. For the numerical experiment, we consider q reservoirs R_{i_l} such that $i_1 = 2$ and $i_l = \sum_{k=1}^{l-1} g_{i_l} + l + 1$ for $l = 2, \dots, q-1$, and $g_{i_l} = 300$, $l = 1, \dots, q$, for each $q \in \{2, 5, 10, 20, 30, 40, 60\}$. Table 1 displays the obtains results. We report the order of the original system (n), the order of the reduced system (r), and the CPU time in second (tcpu), of each method. In this table we also report the average simulation time of the original system (tso), the average simulation time of the reduced system (tsr) and the average error

$$e_* = \frac{\|y(t) - \hat{y}(t)\|}{\|y(t)\|},$$

for 10 inputs $u(t)$ randomly generated with $t \in [0, 100]$, where the norm $\|\cdot\|$ is defined as

$$\|h(t)\| = \left(\int_0^{100} h(t)^2\, dt\right)^{1/2} \approx \texttt{sqrt(trapz(t,h.\^{}2))},$$

the function `trapz(t,h.^2)` computes an approximation of the integral of $h(t)^2$ with respect to t using trapezoidal integration, $\texttt{h}=(h(t_0), \ldots, h(t_{100}))$, and $\texttt{t}=(t_0, t_1, \ldots, t_{100})$ with $t_k = t_{k-1} + 0.1k$, $t_0 = 0$. The described notation is also used in the remaining tables.

		`matlab_bt`				`ra_srbt`			
n	tso	r	tcpu	tsr	e_*	r	tcpu	tsr	e_*
603	0.64	1	4.09	0.01	2.1e-14	1	2.50	0.00	3.5e-14
1506	6.49	1	51.25	0.01	5.1e-14	1	12.68	0.01	1.1e-13
3011	38.52	1	425.98	0.00	1.2e-13	1	47.80	0.00	1.3e-13
6021	265.92	1	3860.78	0.00	2.7e-13	2	313.11	0.02	2.7e-13
9031	907.56	1	12443.45	0.01	3.7e-13	1	596.31	0.01	3.7e-13

Tab. 1: Results for the system of n water reservoirs

For our second numerical example we chose the compartmental networks of n components (see [24] for details) where the matrix $A = (a_{ij})$ is given by

$$a_{ij} = \begin{cases} -\left(\sum_{j \neq i} k_{ji} + k_{oi}\right), & i = j, \\ k_{ij}, & i \neq j, \end{cases} \quad \text{with } k_{ii} = 0,$$

$x_i(t)$ is the quantity (or, concentration) of material involved in compartment i at time t, $k_{ij}x_j(t)$ is the mass flow from compartment j to i, $k_{oi}x_i(t)$ is the sum of all outflows of compartment i. Further, the external inflow of compartment i is given by $\sum_{j=1}^{m} b_{ij}u_j(t)$ where $u_j(t)$ is the jth input resource.

For illustration, we construct the compartmental network with n components shown in Figure 2. For the numerical experiment, we consider a random compartmental network in such a way that k_{ij} is uniformly distributed on the interval $(0, 1)$, the compartments C_1 and C_2 have the same input source, i.e., $B = (1\ 1\ 0 \cdots 0)^T \in \mathbb{R}^{n\times 1}$, $C = (1\ 1 \cdots 1) \in \mathbb{R}^{1\times n}$, and $D = 0$. Table 2 displays the obtains results.

For our last example we consider the partial differential equation (PDE) (see [25]),

$$\frac{\partial x}{\partial t} = \frac{\partial^2 x}{\partial z^2} + \frac{\partial^2 v}{\partial v^2} + 20\frac{\partial x}{\partial z} - 180x + f(z, v)u(t),$$

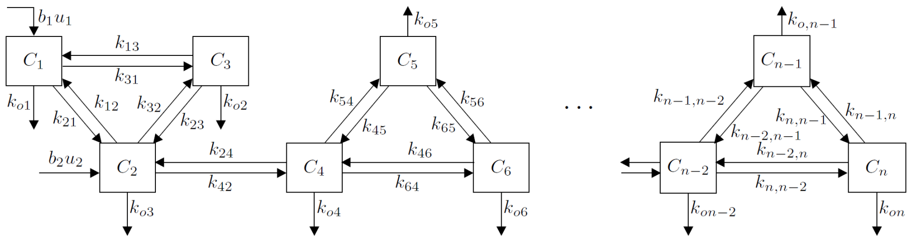

Fig. 2: Compartmental network of n components

		matlab_bt				ra_srbt			
n	tso	r	tcpu	tsr	e_*	r	tcpu	tsr	e_*
600	1.15	5	5.21	0.00	3.0e-09	5	2.48	0.00	3.0e-09
1500	5.98	4	48.91	0.00	3.3e-09	4	12.75	0.01	3.3e-09
3000	31.11	5	395.63	0.00	3.0e-10	5	52.56	0.00	3.0e-10

Tab. 2: Results for the compartmental network of n components

where x is a function of the time (t), horizontal position (z) and vertical position (v). The boundaries of interest in this problem lie on a square with oppositive corners at $(0,0)$ and $(1,1)$. The function $x(t,z,v)$ is zero on these boundaries. This PDE can be discretized with centered difference approximations on a grid $N_z \times N_v$. A state-space equation (1) of order $n = N_zN_v$ results from the discretization. The input vector $B \in \mathbb{R}^{n\times 1}$ of the system corresponds to $f(z,v)$ and is composed of random elements. The output vector $C \in \mathbb{R}^{1\times n}$ of the system is equated to the input vector for simplicity, and $D = 0$. For the numerical experience, we take $N_z = N_v = N \in \{10, 20, 30, 40, 50, 60, 80, 100\}$. The results for this example are summarized in Table 3.

We observed that `ra_srbt` has better performance than `matlab_bt` for the water reservoirs and the compartmental networks. But, for PDE example, the results of Table 3 indicate that `matlab_bt` has better performance than the new approach. Such results are consequence of the ill-conditioning of the Lyapunov equations associated with the gramians W_c and W_o. This ill-conditioning negatively influences the behavior of `ra_lyap` method, which requires, in this case, a larger number of iterations and function evaluations, furthermore, `ra_lyap` gets poor approximations of the gramians, which yields a final r inappropriate.

		matlab_bt				ra_srbt			
n	tso	r	tcpu	tsr	e_*	r	tcpu	tsr	e_*
100	0.02	5	0.06	0.00	8.3e-08	5	0.12	0.00	8.3e-08
400	0.29	7	1.33	0.00	4.9e-08	6	3.01	0.01	5.8e-07
900	2.85	8	9.42	0.00	4.7e-08	6	20.98	0.00	3.7e-06
1600	12.71	9	46.85	0.04	3.1e-08	6	83.35	0.01	1.0e-05
2500	42.37	10	165.99	0.01	1.3e-08	6	264.08	0.00	3.5e-05
3600	124.76	10	527.41	0.01	3.0e-08	6	654.77	0.00	2.5e-05

Tab. 3: Results for the partial differential equation

5 Final Remarks

In this paper, a residual approach of the square root balanced truncation algorithm for model order reduction of continuous, linear and time-invariante compartmental systems is proposed. The new approach uses the `ra_lyap` method to approximate the controllability and observability gramians in the square root balanced truncation algorithm, that is the step that produces a high computational cost in this algorithm of balanced truncation. In the numerical examples we note that the CPU time required to compute the Cholesky factorization or the SVD is significantly less than the time of resolution of the Lyapunov equations. For example, for problems with dimension $n \approx 3000$, the CPU time of the Cholesky factorization is about 0.22 seconds, and the CPU time of the SVD is approximately 0.21 seconds, but the CPU time of the resolution of one Lyapunov equation is about 26.42 seconds. We cannot guarantee that this fact always happen for any linear and time-invariant system, but we believe that it is true for linear and time-invariant compartmental systems, whose verification is an open problem.

The preliminary numerical results show that `ra_srbt` performs efficiently for the model order reduction of linear and time-invariant compartmental systems. Due to its simplicity, `ra_srbt` is very easy to implement, and, furthermore as `ra_lyap` is a residual algorithm of low storage memory that does not requires of large matrix factorizations, the use `ra_srbt` for the model order reduction of linear and time-invariant compartmental systems is attractive. Of course, when the problem is ill-conditioned (e.g., PDE example) is convenient to combine `ra_lyap` with a modern preconditioning strategy to improve their performance (see, for example, Chehab and Raydan [26]). The design of preconditioning strategies for `ra_lyap` is a topic to develop in future research.

Acknowledgements: We would like to thank two anonymous referees for their helpful suggestions which improved the quality of this paper.

References

[1] Jacquez J.A. *Modeling with Compartments.* Biomedware, Ann Arbor, 1999.

[2] Davison E.J. A method for simplifying linear dynamic systems. *IEEE Transaction on Automated Control*, 11(1):93–101, 1966.

[3] Shamash Y. Model reduction using the routh stability criterion and the padé approximation technique. *Int. J. Control*, 21(3):475–489, 1975.

[4] Shamash Y. Linear system reduction using padé approximation to allow retention of dominant models. *Int. J. Control*, 21(2):257–272, 1975.

[5] Guyan R.J. Reduction of stiffness and mass matrices. *AIAA Journal*, 3(2):380–386, 1965.

[6] De Villemagne C. and Skelton R.E. Model reductions using a projection formulation. *Int. J. Control*, 46:2141–2169, 1987.

[7] Freund R.W. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123:395–421, 2000.

[8] Freund R.W. *Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation*, volume 1, pages 435–498. B.N. Datta, Birkhäuser, Boston, 1999.

[9] Moore B.C. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, 26:17–32, 1981.

[10] Laub A.J., Heath M.T., Paige C.C., and Ward R.C. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Automat. Control*, 32:115–122, 1987.

[11] Tombs M. and Postlethweite I. Truncated balanced realization of a stable non-minimal state-space system. *Int. J. Control*, 46:1319–1330, 1987.

[12] Safonov M.G. and Chiang R.Y. A schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, 34:729–733, 1989.

[13] Varga A. Efficient minimal realization procedure based on balancing. In Tzafestas S.G. EL Moudni A., Borne P., editor, *Proceedings of the IMACS/IFAC Symposium on Modelling and Control of Technological Systems*, volume 2, pages 42–47, 1991.

[14] Pernebo L. and Silverman L.M. Model reduction via balanced state space representation. *IEEE Trans. Automat. Control*, AC-27(2):382–382, 1982.

[15] Enns D. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Proceedings of the 23^{rd} IEEE Conf. Decision and Control*, 1984.

[16] La Cruz W. Residual spectral algorithm for solving monotone equations on a hilbert space. *Applied Mathematics and Computation*, 219:6633–6644, 2013.

[17] La Cruz W. and Raydan M. Nonmonotone spectral methods for large-scale nonlinear systems. *Optimization Methods & Software*, 18:583–599, 2003.

[18] La Cruz W., Martínez J.M., and Raydan M. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of Computation*, 75:1429–1448, 2006.

[19] Antoulas A.C. *Approximation of large-scale Dynamical Systems. Advances in Design and Control.* SIAM, Philadelphia, 2005.

[20] Zhou K., Doyle J., and Glover K. *Robust and Optimal Control.* Prentice-Hall, Upper Saddle River, NJ, 1996.

[21] Bartels R.H. and Stewart G.W. Solution of the matrix equation $Ax + Bx = C$. *Communications of the ACM*, 15:820–826, 1972.

[22] Barzilai J. and Borwein J. Two point step gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[23] Reis T. and Virnik E. Positivity preserving balanced truncation for descriptor systems. *SIAM Journal on Control and Optimization*, 48:2600–2619, 2009.

[24] Godfrey K. *Compartmental Models and Their Applications.* Academic Press, London, 1983.

[25] Chahlaoui Y. and Van Dooren P. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. Technical report, U.S. Geological Survey, February 2002. SLICOT Working Note 2002-2.

[26] Chehab J. and Raydan M. An implicit preconditioning strategy for large-scale generalized sylvester equations. *Applied Mathematics and Computation*, 217:8793–8803, 2011.