Raport Badawczy

Research Report

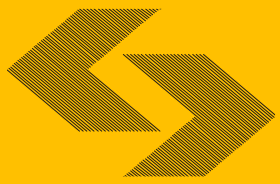
RB/86/2009

Iterative-interpolation algorithms for L_2 model reduction

W. Krajewski, U. Viaro

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



Control and Cybernetics

vol. **38** (2009) No. 2

Iterative-interpolation algorithms for L_2 model reduction*

by

Wiesław Krajewski[1] and Umberto Viaro[2]

[1]Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland

[2]Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica
Università di Udine
via delle Scienze 208, 33100 Udine, Italy

e-mail: krajewsk@ibspan.waw.pl, viaro@uniud.it

Abstract: This paper is concerned with the construction of reduced–order models for high–order linear systems in such a way that the L_2 norm of the impulse–response error is minimized. Two convergent algorithms that draw on previous procedures presented by the same authors, are suggested: one refers to s–domain representations, the other to time–domain state–space representations. The algorithms are based on an iterative scheme that, at any step, satisfies certain interpolation constraints deriving from the optimality conditions. To make the algorithms suitable to the reduction of very large–scale systems, resort is made to Krylov subspaces and Arnoldi's method. The performance of the reduction algorithms is tested on two benchmark examples.

Keywords: linear systems, model reduction, output–error minimization, L_2 norm, Krylov subspaces, Arnoldi's algorithm.

1. Introduction

Linear time–invariant models are often used to describe phenomena in physical and economic contexts because many tools are available for their study. However, when the system complexity is high or its size is big, the number of state variables may be too large for simulation and control purposes, and it is mandatory to approximate the original high–order model by means of a lower–order one.

Many approaches to model reduction have been proposed over the past decades. Among the most popular ones, we can mention the Padé-like methods leading to the retention of some Markov parameters and time moments (Bultheel

*Submitted: April 2007; Accepted: January 2009.

and Van Barel, 1986), the aggregation techniques retaining some modes of the original system (Hickin and Sinha, 1980), the Hankel–norm optimization methods (Glover, 1984), the techniques based on principal component analysis and balanced realizations (Moore, 1981), and the methods aiming at the retention of suitable first–order information indices (Markov parameters or time moments) and second–order information indices (impulse–response powers) (de Villemagne and Skelton, 1987).

Recently, the techniques based on moment matching have been tackled with the aid of numerically robust and reliable procedures such as the Arnoldi, Lanczos or rational Krylov methods (see, for example, Gallivan, Grimme and Van Dooren, 1996; Datta, 2003 and Antoulas, 2005), thus allowing the reduction of very large–scale systems. On the other hand, the techniques based on the minimization of an error norm still seem to be more appropriate to model reduction, which is essentially an optimization problem.

This paper deals with the minimization of the L_2 norm of the output error, defined as the difference between the impulse responses of the original and reduced–order model. The L_2 norm has an appealing physical interpretation (power) which explains its wide-spread use (see, for example, Meier and Luenberger, 1967; Wilson, 1974; Hyland and Bernstein, 1985 concerning the time domain, and Spanos, Milman and Mingori, 1992; Krajewski et al., 1995; Fulcheri and Olivi, 1998 concerning the frequency domain).

Finding the L_2–optimal reduced–order model of a complex system is a hard task that often requires the solution of an ill–conditioned mathematical programming problem. Many of the available methods are therefore difficult to implement and rather inefficient; in particular, gradient techniques are not always satisfactory. Among the non–gradient approaches, the iterative–interpolation algorithm first suggested in Lepschy et al. (1991) and further developed in Krajewski et al. (1995) and Ferrante et al. (1998) with reference to the *frequency domain* seems to be one of the most efficient procedures. This algorithm, however, is not always convergent. To overcome this difficulty, a variant of the method characterized by steps of shortened length has been proposed in Ferrante et al. (1999). The iterative–interpolation approach has recently been extended to *time–domain state–space* representations (Gugercin, Antoulas and Beattie, 2006). In this paper we pursue the ideas from Ferrante et al. (1999) and Gugercin, Antoulas and Beattie (2006) by suggesting two convergent algorithms in frequency and time domain, respectively

The next part of the paper is organized as follows. Section 2 formulates the L_2–optimal model–reduction problem and recalls the iterative interpolation method. Section 3 shows how convergence can be guaranteed and proposes two algorithms for models in either input–output or state–space form. These algorithms are applied in Section 4 to two benchmark examples that show the efficiency of the proposed techniques.

2. Problem statement and iterative–interpolation method

Let the state–space equations of the original full–order linear time–invariant *asymptotically–stable* system be

$$\dot{x}(t) = A\,x(t) + b\,u(t)\,, \tag{1}$$
$$y(t) = c\,x(t)\,, \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, and n is the order of a minimal realization of the system. The input–output behavior of system (1)–(2) is characterized by the transfer function

$$f(s) = \frac{n(s)}{d(s)} = c\,(s\,I - A)^{-1}\,b\,, \tag{3}$$

where $\deg d(s) = n$ and $\deg n(s) \leq n-1$. Similarly, assume that the state–space equations of the reduced–order model are:

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{b}u(t)\,, \tag{4}$$
$$\tilde{y}(t) = \tilde{c}\tilde{x}(t)\,, \tag{5}$$

where $\tilde{x}(t) \in \mathbb{R}^{\tilde{n}}$ with $\tilde{n} < n$, $\tilde{y}(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$. The related transfer function is

$$g(s) = \frac{m(s)}{c(s)} = \tilde{c}\,(s\,I - \tilde{A})^{-1}\,\tilde{b}\,, \tag{6}$$

where $\deg c(s) = \tilde{n}$ and $\deg m(s) \leq \tilde{n} - 1$.

In frequency domain, the optimal model–reduction problem can be formulated as follows: given an original transfer function $f(s)$ of order n, find a transfer function $g(s)$ of preassigned lower order $\tilde{n} < n$ in such a way that the (squared) H_2 norm of $e(s) := f(s) - g(s)$, that is,

$$\|e\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} e(j\omega)e^*(j\omega)d\omega\,, \tag{7}$$

is minimized (the star indicates complex conjugate).

The time–domain state–space version of this problem can be formulated as follows: given a state–space model (A, b, c) of minimal dimension n, find a model $(\tilde{A}, \tilde{b}, \tilde{c})$ of dimension $\tilde{n} < n$ so as to minimize the (squared) L_2 norm of the output error $e(t) := y(t) - \tilde{y}(t)$, that is,

$$\|e\|_2^2 = \int_{-\infty}^{\infty} e^2(t)\,d\,t\,, \tag{8}$$

where $y(t)$ and $\tilde{y}(t)$ are the responses of the full–order and reduced–order model, respectively, to a unit–impulse input $u(t)$.

Assuming for notational simplicity that all of the poles of $g(s)$ are simple (but not necessarily real) and denoting them by $p_1, \ldots, p_{\tilde{n}}$, function $g(s)$ minimizes (7) only if the following *interpolation conditions* are satisfied (Meier and Luenberger, 1967):

$$\left.\begin{array}{rcl} f(-p_k) - g(-p_k) & = & 0\,, \\ f'(-p_k) - g'(-p_k) & = & 0\,, \end{array}\right\} \quad k = 1, 2, \ldots, \tilde{n}\,. \tag{9}$$

These conditions can be expressed more compactly in terms of the numerator and denominator polynomials in (3) and (6) as

$$n(s)\, c(s) - m(s)\, d(s) = q(s)\, c^2(-s)\,, \tag{10}$$

where $q(s)$ is a polynomial whose degree is, at the most, $n - \tilde{n} - 1$.

Equating the coefficients of the equal powers of s on both sides of the polynomial identity (10), we obtain $n + \tilde{n} - 1$ equations in the $2\,\tilde{n}$ unknown coefficients of $m(s)$ and $c(s)$ as well as in the $n - \tilde{n} - 1$ unknown coefficients of the "auxiliary" polynomial $q(s)$. These equations can be solved by means of an iterative procedure based on the recurrence relation

$$n(s)\, c^{(h+1)}(s) - m^{(h+1)}\,(s) d(s) \; = q^{(h+1)}(s)\, [c^{(h)}(-s)]^2\,, \tag{11}$$

which determines polynomials $c^{(h+1)}(s)$ and $m^{(h+1)}$ from the polynomial $c^{(h)}(s)$ previously computed.

The resulting algorithm can be presented as follows:

Algorithm 1

1. Make a guess of the initial reduced–order denominator polynomial $c^{(0)}(s)$ and set $h = 0$.
2. Given $c^{(h)}(s)$, evaluate $c^{(h+1)}(s)$, $m^{(h+1)}(s)$ and $q^{(h+1)}(s)$ on the basis of (11).
3. If the stopping criterion is satisfied, form the reduced–order denominator and numerator as $c(s) = c^{(h+1)}(s)$ and $m(s) = m^{(h+1)}(s)$, respectively. Otherwise, set $h = h + 1$ and return to Step 2.

Step 2 requires the solution of a set of linear equations. The corresponding system matrix is sparse with a structure that guarantees the efficiency of the procedure, whose details are illustrated in Krajewski et al. (1995). Equation (11) can be given an interesting interpretation; precisely, it states that function $g^{(h+1)}(s)$ must interpolate the original transfer function $f(s)$, with intersection number 2, at the opposites of the poles $p_1^{(h)}, \ldots, p_{\tilde{n}}^{(h)}$ of $g^{(h)}(s)$, that is,

$$\left.\begin{array}{rcl} f(-p_k^{(h)}) - g^{(h+1)}(-p_k^{(h)}) & = & 0\,, \\ f'(-p_k^{(h)}) - (g^{(h+1)})'(-p_k^{(h)}) & = & 0\,, \end{array}\right\} \quad k = 1, 2, \ldots, \tilde{n}\,. \tag{12}$$

In other words, the numerator of the current error function $e^{(h+1)}(s) := f(s) - g^{(h+1)}(s)$ must contain the factor $[c^{(h)}(-s)]^2$. This is the reason why the above procedure has been called *iterative–interpolation* algorithm.

The idea of iterative interpolation has been recently extended to state–space representations such as (4) and (5) in Gugercin, Antoulas and Beattie (2006), where the interpolation problem arising at each iteration is solved by means of Krylov–subspace methods. To this purpose, denoting by $p_k^{(h)}$ the $\tilde{n}$ eigenvalues $\lambda_k(\tilde{A}^{(h)})$, $k = 1, 2, \ldots, \tilde{n}$, of the reduced–order system matrix $\tilde{A}^{(h)}$ computed in the preceding iteration, two matrices V and Z are formed in the current iteration such that

$$Im(V) = span\{(-p_1^{(h)} I - A)^{-1} b, \ldots, (-p_{\tilde{n}}^{(h)} I - A)^{-1} b)\} \,, \tag{13}$$

$$Im(Z) = span\{(-p_1^{(h)} I - A^T)^{-1} c^T, \ldots, (-p_{\tilde{n}}^{(h)} I - A^T)^{-1} c^T)\} \tag{14}$$

and $Z^T V = I$. In this way the reduced–order model (4) and (5) with

$$\tilde{A} = \tilde{A}^{(h+1)} := Z^T A V \;,\; \tilde{b} = \tilde{b}^{(h+1)} := Z^T b \;,\; \tilde{c} = \tilde{c}^{(h+1)} := c V \tag{15}$$

satisfies the following interpolation conditions analogous to (12) (Grimme, 1997; Antoulas, 2005):

$$\left.\begin{aligned} c(-p_k^{(h)} I - A)^{-1} b &= \tilde{c}^{(h+1)} (-p_k^{(h)} I - \tilde{A}^{(h+1)})^{-1} \tilde{b}^{(h+1)} \;, \\ c(-p_k^{(h)} I - A)^{-2} b &= \tilde{c}^{(h+1)} (-p_k^{(h)} I - \tilde{A}^{(h+1)})^{-2} \tilde{b}^{(h+1)} \;, \end{aligned}\right\} \quad k = 1, 2, \ldots, \tilde{n} \,. \tag{16}$$

The corresponding algorithm can be presented as follows:

Algorithm 2

1. Make a guess of the eigenvalues $p_1^{(0)}, \ldots, p_{\tilde{n}}^{(0)}$ of the initial reduced–order system matrix $\tilde{A}^{(0)}$ (often, the $\tilde{n}$ dominant eigenvalues of A represent a good starting point) and set $h = 0$.
2. Choose V and Z such that
   $Im(V) = span\,\{(-p_1^{(h)} I - A)^{-1}\, b, \ldots, (-p_{\tilde{n}}^{(h)} I - A)^{-1}\, b\}$,
   $Im(Z) = span\,\{(-p_1^{(h)} I - A^T)^{-1}\, c^T, \ldots, (-p_{\tilde{n}}^{(h)} I - A^T)^{-1}\, c^T\}$,
   $Z = Z\,(Z^T V)^{-T}$.
3. Compute $\tilde{A}^{(h+1)} = Z^T A V$.
4. If the stopping criterion is satisfied, form the reduced–order model matrices as $\tilde{A} = \tilde{A}^{(h+1)}$, $\tilde{b} = Z^T b$ and $\tilde{c} = c V$. Otherwise, let $p_k^{(h+1)} = \lambda_k(\tilde{A}^{(h+1)})$, $k = 1, \ldots, \tilde{n}$, set $h = h + 1$ and return to phase 2.

To find V and Z, the rational Arnoldi method can conveniently be applied (Gugercin, Antoulas and Beattie, 2006), which makes the above algorithm suitable to the reduction of very large–scale systems.

Algorithms 1 and 2 have proven to be efficient (Krajewski et al., 1995; Gugercin, Antoulas and Beattie, 2006), but are not always convergent. The next section shows how these algorithm can be modified to ensure convergence.

3. Convergent variants of the iterative–interpolation algorithms

Denoting by $\mathbf{c}^{(i)}$ the vector formed from the coefficients of $s^{\tilde{n}-1}$, $s^{\tilde{n}-2}$, ..., s^0, in every *monic* polynomial $c^{(i)}(s)$ generated by (11), recursion (11) can be reformulated as

$$\mathbf{c}^{(h+1)} = \Phi(\mathbf{c}^{(h)})\,, \tag{17}$$

where $\Phi : \mathbb{R}^{\tilde{n}} \to \mathbb{R}^{\tilde{n}}$ is a continuously differentiable function. In this way, the reduced–order model minimizing (7) corresponds to a fixed point $\hat{\mathbf{c}}$ of Φ:

$$\hat{\mathbf{c}} = \Phi(\hat{\mathbf{c}})\,. \tag{18}$$

It has been shown in Krajewski et al. (1995) and Ferrante et al. (1999) that:
(i) the eigenvalues of the Jacobian $\frac{\partial \Phi}{\partial \mathbf{c}}$ at every fixed point $\hat{\mathbf{c}}$ are real,
(ii) at a fixed point $\hat{\mathbf{c}}$ corresponding to a saddle point of the objective function at least one eigenvalue of the Jacobian is greater than 1, and
(iii) at a fixed point corresponding to a minimum of the objective function every eigenvalue of the Jacobian is less than 1 but not necessarily greater than -1, which explains why the iterative–interpolation algorithm does not always converge.

To avoid this difficulty, it has been suggested in Ferrante et al. (1999) to replace Φ by another continuously differentiable function with the following properties:
(i) it has the same fixed points $\hat{\mathbf{c}}$ as Φ,
(ii) all the eigenvalues of its Jacobian at every $\hat{\mathbf{c}}$ have magnitude less than 1, and
(iii) it can easily be obtained from Φ with no *a priori* information about $\hat{\mathbf{c}}$.

These properties are satisfied for a suitably small value of the real parameter α by the function $\Phi_\alpha : \mathbb{R}^{\tilde{n}} \to \mathbb{R}^{\tilde{n}}$ obtainable from Φ according to

$$\Phi_\alpha(\mathbf{c}) = \alpha\Phi(\mathbf{c}) + (1-\alpha)\mathbf{c}\,. \tag{19}$$

Therefore, the iterative procedure (17) can be replaced by

$$\bar{\mathbf{c}}^{(h+1)} = \Phi_\alpha(\bar{\mathbf{c}}^{(h)})\,. \tag{20}$$

Starting from $\bar{\mathbf{c}}^{(h)} = \mathbf{c}^{(h)}$, recursion (20) determines vector $\bar{\mathbf{c}}^{(h+1)}$ as a linear combination of $\mathbf{c}^{(h)}$ and $\mathbf{c}^{(h+1)} = \Phi(\mathbf{c}^{(h)})$ according to the combination coefficients $1-\alpha$ and α, respectively.

Denoting by λ_m the smallest eigenvalue of $\frac{\partial \Phi}{\partial \mathbf{c}}(\hat{\mathbf{c}})$, if

$$0 < \alpha < \alpha_m = \frac{2}{1 - \lambda_m}\,, \tag{21}$$

then all the eigenvalues of $\frac{\partial \Phi_\alpha}{\partial \mathbf{c}}(\hat{\mathbf{c}})$ have magnitude less than 1 (Ferrante et al., 1999), and the sequence of vectors $\{\bar{\mathbf{c}}^{(i)}\}$ generated by (20) converges to $\hat{\mathbf{c}}$ from a suitable neighbourhood of $\hat{\mathbf{c}}$.

From the above arguments the following algorithm is obtained:

Algorithm 3

1. Select $\alpha \in (0,1)$, make a guess of the initial coefficient vector $\bar{\mathbf{c}}^{(0)}$, and set $h = 0$.
2. Let $\mathbf{c}^{(h)} = \bar{\mathbf{c}}^{(h)}$ and compute $\mathbf{c}^{(h+1)} = \Phi(\mathbf{c}^{(h)})$ on the basis of (11) as in Algorithm 1.
3. Form $\bar{\mathbf{c}}^{(h+1)} = \alpha\, \mathbf{c}^{(h+1)} + (1-\alpha)\, \bar{\mathbf{c}}^{(h)}$.
4. If the stopping criterion is satisfied, form the reduced–order denominator and numerator as $c(s) = c^{(h+1)}(s)$ and $m(s) = m^{(h+1)}(s)$, respectively. Otherwise, set $h = h + 1$ and return to phase 2.

It follows that the steps made by Algorithm 3 at each iteration are α times shorter than the steps made by Algorithm 1.

A similar approach can be followed to ensure the convergence of a variant of the time–domain state–space iterative–interpolation procedure. It suffices to introduce a few additional operations into Algorithm 2 for: (i) computing the characteristic polynomial $c^{(h+1)}(s)$ associated with $\tilde{A}^{(h+1)}$, (ii) shortening the step made at every iteration by a factor α, and (iii) evaluating the roots $\bar{p}_1, \ldots, \bar{p}_{\tilde{n}}$ of the polynomial $\bar{c}^{(h+1)}(s)$ formed from the vector $\bar{\mathbf{c}}^{(h+1)}$.

The resulting algorithm is outlined next:

Algorithm 4

1. Select $\alpha \in (0,1)$, and form the coefficient vector $\bar{\mathbf{c}}^{(0)}$ by choosing the roots $\bar{p}_1^{(0)}, \ldots, \bar{p}_{\tilde{n}}^{(0)}$ of the characteristic polynomial $\bar{c}^{(0)}(s)$ associated with the initial reduced–order system matrix $\tilde{A}^{(0)}$. Set $h = 0$.
2. Choose V and Z such that
   $Im(V) = span\,\{(-\bar{p}_1^{(h)} I - A)^{-1}\, b, \ldots, (-\bar{p}_{\tilde{n}}^{(h)} I - A)^{-1}\, b\}$,
   $Im(Z) = span\,\{(-\bar{p}_1^{(h)} I - A^T)^{-1}\, c^T, \ldots, (-\bar{p}_{\tilde{n}}^{(h)} I - A^T)^{-1}\, c^T\}$,
   $Z = Z\,(Z^T V)^{-T}$.
3. Compute $\tilde{A}^{(h+1)} = Z^T\, A\, V$.
4. If the stopping criterion is satisfied, form the reduced–order model matrices as $\tilde{A} = \tilde{A}^{(h+1)}$, $\tilde{b} = Z^T\, b$ and $\tilde{c} = c\, V$. Otherwise:
   (i) find the characteristic polynomial $c^{(h+1)}(s)$ associated with $\tilde{A}^{(h+1)}$ and form the related coefficient vector $\mathbf{c}^{(h+1)}$,

(ii) compute $\bar{\mathbf{c}}^{(h+1)} = \alpha\,\mathbf{c}^{(h+1)} + (1-\alpha)\,\bar{\mathbf{c}}^{(h)}$ and form the related polynomial $\bar{c}^{(h+1)}(s)$,
(iii) evaluate the roots $\bar{p}_k^{(h+1)}$, $k = 1, \ldots, \tilde{n}$, of $\bar{c}^{(h+1)}(s)$, and
(iv) set $h = h + 1$ and return to phase 2.

Shortening the steps by α ensures the convergence of Algorithms 3 and 4 also when Algorithms 1 and 2 do not converge. However, if the latter algorithms converge, which is not known *a priori*, their modified versions may require a larger number of iterations to find a (locally) optimal solution, as shown by the examples in the next section.

REMARK As already pointed out at the beginning of this section, the previous algorithms for L_2–optimal model–reduction can be considered as iterative schemes for finding the fixed points of the mapping Φ in (17) and (18), implicitly defined by the optimality conditions (10). The determination of fixed points using suitable contractive mappings is a classical topic in mathematical analysis and has attracted continuous attention over the years. Among the methods for fixed point search, the iterative scheme proposed by Mann in 1953 and the one developed by Ishikawa in 1974 seem to be the most popular ones (Rhoades, 1976); their behaviour for different classes of nonlinear mappings is still being studied extensively (see, for example, Miao, 2007, and Arandelovic, 2009). Using the above notation, Mann's iteration can be written as

$$\mathbf{c}^{(h+1)} = \alpha_h\,\Phi(\mathbf{c}^{(h)}) + (1-\alpha_h)\,\mathbf{c}^{(h)}\,,\quad h \geq 0\,, \tag{22}$$

where $0 \leq \alpha_h \leq 1$ for every $h \geq 0$, and Ishikawa's iteration as

$$\mathbf{c}^{(h+1)} = \alpha_h\,\Phi\big(\beta_h\,\Phi(\mathbf{c}^{(h)}) + (1-\beta_h)\,\mathbf{c}^{(h)}\big) + (1-\alpha_h)\,\mathbf{c}^{(h)}\,,\quad h \geq 0\,, \tag{23}$$

where $0 \leq \alpha_h \leq \beta_h \leq 1$ for every $h \geq 0$. Therefore, (22) is a special case of (23). The inspection of (17), (19) and (20) shows that Algorithm 3 corresponds to Mann's iteration scheme.

4. Examples

In the following, the modified iterative–interpolation approach is applied to two meaningful examples. The purpose of this section is threefold, that is: (i) to show that the modified algorithms can find an optimal solution also when the original iterative–interpolation algorithms fail, (ii) to see how the convergence of the modified algorithms depends on α, and (iii) to compare the speed of the modified algorithms with that of the original ones when the latter converge.

4.1. Model of a flexible structure

The full–order state–space model of a flexible structure is described by the following matrices (Gawronski and Juang, 1990):

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -5.4545 & 4.5455 & 0 & -0.0545 & 0.0455 & 0 \\ 10 & -21 & 11 & 0.10 & -0.2100 & 0.1100 \\ 0 & 5.5 & -6.5 & 0 & 0.0550 & -0.0650 \end{bmatrix},$$

$$b^T = \begin{bmatrix} 0 & 0 & 0 & 0.0909 & 0.4 & -0.5 \end{bmatrix}, \quad c = \begin{bmatrix} 2 & -2 & 3 & 0 & 0 & 0 \end{bmatrix}.$$

The characteristic equation has three pairs of complex conjugate poles, that is, $-0.0038 \pm \jmath\, 0.8738$, $-0.0297 \pm \jmath\, 2.4374$ and $-0.1313 \pm \jmath\, 5.1217$, which hinders the construction of odd–order approximants. In fact, neither Algorithm 1 nor Algorithm 2 allows us to determine an optimal reduced–order model of order 1, 3 or 5.

Instead, the iterative–interpolation Algorithms 3 and 4 with suitably shortened steps can find these odd–order approximants in a small number of iterations starting from the same initial guesses (and stringent stopping criterion). In particular, the optimal first–order model turns out to be:

$$\dot{\tilde{x}}(t) = -0.6746\tilde{x}(t) - 0.1948u(t)\,,$$
$$\tilde{y}(t) = 1.8899\tilde{x}(t)\,.$$

The value of the squared error norm of the related output error is $\|e\|_2^2 = 3.9758$. The number of iterations required to find this model is shown in Table 1 for different values of α.

Table 1. Number of iterations vs. α for the optimal first–order approximant

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
number of iterations	54	24	14	9	6	11	20	41	184	∞

In the case of even–order approximants, the original iterative–interpolation Algorithms 1 and 2 converge too. For example, starting from $p_1^{(0)} = -1$ and $p_2^{(0)} = -2$, Algorithm 2 arrives at the following optimal second-order solution in 6 iterations only:

$$\tilde{A} = \begin{bmatrix} -0.0063 & 0.7635 \\ -0.9999 & -0.0013 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} -0.1258 \\ 0.0003 \end{bmatrix}, \quad \tilde{c} = \begin{bmatrix} 0.0233 & -1.6646 \end{bmatrix}.$$

The value of the squared error norm of the related output error is $\|e\|_2^2 = 0.2934$.

Algorithm 4 requires a greater number of iterations to find this model. Table 2 shows how this number depends on α .

Table 2. Number of iterations vs. α for the optimal second–order approximant

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
number of iterations	125	84	52	36	28	21	17	13	10	6

4.2. Model of a CD player

The full–order state–space model describing the dynamic relation between the lens actuator and the radial–arm position of a portable CD player is characterized by 120 state variables (Chahlaoui and Van Dooren, 2002). Since all of the poles of this system are complex (sixty complex conjugate pairs), it is again difficult to find an optimal approximant of odd order. For example, starting from the (arbitrary) initial guess $p_1^0 = -0.5$, $p_2^0 = -1$, $p_3^0 = -2$, $p_4^0 = -3$, $p_5^0 = -4$, $p_6^0 = -5$, $p_7^0 = -6$, $p_8^0 = -7$, $p_9^0 = -8$, Algorithm 2 can not find a 9th–order model. Instead, Algorithm 4 with $\alpha = 0.1$ finds an optimal solution of order 9 in 55 iterations starting from the same initial guess. The square norm of the related output error is $\|e\|_2^2 = 30.2335$. Note, by comparison, that the 9th–order model obtained according to the popular TBR (Truncated Balanced Realization) method (Moore, 1981) is characterized by $\|e\|_2^2 = 35.149$. With $\alpha = 0.5$ Algorithm 4 arrives at the optimal solution in 16 iterations only.

Algorithm 2 finds an optimal 10th–order approximant in 7 iterations starting from $p_1^0 = -0.5$, $p_2^0 = -1$, $p_3^0 = -2$, $p_4^0 = -3$, $p_5^0 = -4$, $p_6^0 = -5$, $p_7^0 = -6$, $p_8^0 = -7$, $p_9^0 = -8$, $p_{10}^0 = -10$. The corresponding square error norm is $\|e\|_2^2 = 21.04$. As expected, Algorithm 4 requires a larger number of iterations to arrive at the same solution. Table 3 shows how this number depends on α.

Table 3. Number of iterations vs. α for the optimal tenth–order approximant

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
number of iterations	48	30	22	22	21	17	15	12	9	7

5. Conclusions

The iterative–interpolation algorithms for L_2–optimal model reduction have proven to be very efficient compared to alternative procedures with the same objective. Two algorithms operating in either the frequency or the time domain

have been presented in Section 3. They always ensure convergence to an optimal solution. Essentially, this result is obtained by shortening the steps made at every iteration by the original algorithms outlined in Section 4. Two benchmark examples have been worked out in Section 4 to show how the algorithms perform and how the speed of convergence depends on the step size.

Acknowledgements

The authors wish to thank the anonymous reviewers for their valuable comments and for drawing their attention to the relevant literature on fixed–point theory.

References

Antoulas, A.C. (2005) *Approximation of Large–Scale Dynamical Systems.* SIAM Book Series: Advances in Design and Control, Philadelphia.

Arandelovic, I.D. (2009) On the convergence of Ishikawa iterates defined by nonlinear quasi–contractions. *Int. J. Math. Anal.* **3**, 603–610.

Bultheel, A. and Van Barel, M. (1986) Padé techniques for model reduction in linear system theory: a survey. *J. Comp. Appl. Math.* **14**, 401–438.

Chahlaoui, Y. and Van Dooren, P. (2002) A collection of benchmark examples for model reduction of linear time–invariant dynamical systems. *SLICOT Working Note*, Feb. 2002 (available at ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN2002-2.ps.Z).

Datta, B.N. (2003) Krylov subspace methods for large–scale matrix problems in control. *Future Generation Computer Systems* **9**, 1253–1263.

de Villemagne, C. and Skelton, R.E. (1987) Model reductions using a projection formulation. *Int. J. Control* **46**, 2141–2169.

Ferrante, A., Krajewski, W., Lepschy, A. and Viaro, U. (1998) A new algorithm for L_2–optimal model reduction. In: S. Banka, S. Domek and Z. Emirsajlow, eds., *Proceedings of the International Symposium on Mathematical Models in Automation and Robotics*, Miedzyzdroje, Poland, 224–229.

Ferrante, A., Krajewski, W., Lepschy, A. and Viaro, U. (1999) Convergent algorithm for L_2 model reduction. *Automatica* **35**, 75–79.

Fulcheri, P. and Olivi, M. (1998) Matrix rational H_2 approximation: a gradient algorithm based on Schur analysis. *SIAM J. Control Optim.* **36**, 2103–2127.

Gallivan, K., Grimme, E. and Van Dooren, P. (1996) A rational Lanczos algorithm for model reduction. *Numerical Algorithms* **12** 33–63.

Gawronski, W. and Juang, J.N. (1990) Model reduction for flexible structures. In: W. Leondes, ed., *Control and Dynamic Systems* **36**, Academic Press, 143–222.

GLOVER, K. (1984) All optimal Hankel–norm approximations of linear multivariable systems and their L_∞ error bounds. *Int. J. Contr.* **39**, 1115–1193.

GRIMME, E.J. (1997) Krylov Projection Methods For Model Reduction. Ph.D. Thesis, ECE Department, University of Illinois, Urbana–Champaign.

GUGERCIN, S., ANTOULAS, A.C. and BEATTIE, C.A. (2006) A rational Krylov iteration for optimal H_2 model reduction. *Proc. 17th Int. Symp. Mathematical Theory of Networks and Systems*, Kyoto, Japan, 1665–1667.

HICKIN, J. and SINHA, N.K. (1980) Model reduction for linear multivariable systems. *IEEE Trans. Automat. Contr.* **AC-25**, 1121–1127.

HYLAND, D.C. and BERNSTEIN, D.S. (1985) The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton and Moore. *IEEE Trans. Automat. Contr.* **AC-30**, 1201–1211.

KRAJEWSKI, W., LEPSCHY, A., REDIVO-ZAGLIA, M. and VIARO, U. (1995) A program for solving the L_2 reduced–order model problem with fixed denominator degree. *Numerical Algorithms* **9**, 355–377.

LEPSCHY, A., MIAN, G.A., PINATO, G. and VIARO, U. (1991) Rational L_2 approximation: a nongradient algorithm. *Proc. 30th Conference on Decision and Control*, Brighton, England, 2321–2324.

MEIER, L. and LUENBERGER, D.G. (1967) Approximation of linear constant systems. *IEEE Trans. Automat. Contr.* **AC-12**, 585–588.

MIAO, Y. (2007) A modification of Mann's iteration scheme. *East Journal on Approximations* **13**, 211–221.

MOORE, B.C. (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Contr.* **AC-26**, 17–32.

RHOADES, B.E. (1976) Comments on two fixed point iteration methods. *J. Math. Anal. Appl.* **56**, 741–750.

SPANOS, J.T., MILMAN, M.H. and MINGORI, D.L. (1992) A new algorithm for L_2 optimal model reduction. *Automatica* **28**, 897–909.

WILSON, D.A. (1974) Model reduction for multivariable systems. *Int. J. Control* **20**, 57–64.