Commun. Comput. Phys.
doi: 10.4208/cicp.271109.150710s

Vol. **9**, No. 3, pp. 756-779
March 2011

Reduction of Linear Systems of ODEs with Optimal Replacement Variables

Alex Solomonoff[1,*] and Wai Sun Don[2]

[1] *Camberville Research Institute, Somerville, MA, USA.*
[2] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*

Received 27 November 2009; Accepted (in revised version) 15 July 2010

Available online 17 September 2010

To the memory of David Gottlieb

Abstract. In this exploratory study, we present a new method of approximating a large system of ODEs by one with fewer equations, while attempting to preserve the essential dynamics of a reduced set of variables of interest. The method has the following key elements: (i) put a (simple, ad-hoc) probability distribution on the phase space of the ODE; (ii) assert that a small set of *replacement variables* are to be unknown linear combinations of the not-of-interest variables, and let the variables of the reduced system consist of the variables-of-interest together with the replacement variables; (iii) find the linear combinations that minimize the difference between the dynamics of the original system and the reduced system. We describe this approach in detail for linear systems of ODEs. Numerical techniques and issues for carrying out the required minimization are presented. Examples of systems of linear ODEs and variable-coefficient linear PDEs are used to demonstrate the method. We show that the resulting approximate reduced system of ODEs gives good approximations to the original system. Finally, some directions for further work are outlined.

AMS subject classifications: 65M06, 65M30, 65M70

Key words: System reduction, optimal replacement variables, resolved variables, optimal prediction.

1 Introduction

The framework of the problem studied in this paper is a system of linear ordinary differential equations (ODEs)

$$z_t = Fz = \begin{pmatrix} F_0 & F_1 \\ F_2 & F_3 \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad t>0, \tag{1.1}$$

*Corresponding author. *Email addresses:* alex.solomonoff@yahoo.com (A. Solomonoff), wsdon@math.hkbu.edu.hk (W.-S. Don)

with appropriate initial conditions $z(t=0)$, and where $z=(x,y)\in \mathrm{R}^{m+n}$ is divided into a set of *resolved* variables $x\in\mathrm{R}^m$ that one wants to observe or calculate, and a set of *unresolved* variables $y\in\mathrm{R}^n$ that one doesn't need to observe, but which the dynamics of the resolved variables x depend on. Furthermore, it may be that $m\ll n$ or even n infinite, in which case it will not be computationally feasible to solve the full system of equations.

The goal of this study is to approximate (model) the dynamics of x in a computationally efficient way, with useful accuracy, without actually including the full set of unresolved quantities y in the modeling.

This is desirable in many situations: for example, many partial differential equations (PDEs), when discretized into a system of ODEs, require millions of degrees of freedom (DOF) to adequately approximate the dynamics. However, most of these DOF are usually of no interest. Examples of such PDEs include weather simulations and many simulations of fluid or aerodynamic flows. In the flow-around-an-aircraft example, an engineer would be mainly interested in bulk features such as the total lift and drag, or average vorticity as a function of time. A flow field detailed enough to actually resolve all of the dynamics would not be needed in many situations. Calculating solutions to these equations can require large amounts of computing resources, and a system reduction method such as the one studied here has the potential to reduce these resource required, or allow the fast solution of more complex problems.

One approach for system reduction has been developed by Chorin et al. [1, 2] and Gottlieb et al. [3], which has been called the *t-system* or *Optimal Prediction*. An overview of other approaches to the problem can be found in [4].

Much previous work, including much work by Chorin and associates, has revolved around approximate systems having only x as the state variables and eliminating the unresolved variables y. So then x has to somehow represent the dynamics of both the resolved and unresolved variables. Any information about the unresolved dynamics in the t-system must be the result of resolved-dynamics information being discarded or changed. This seems like a limited approach, that could not achieve high accuracy. We propose adding a set of "replacement variables" to the resolved variables to contain the unresolved-dynamics information. This set of variables would be smaller than the actual number of unresolved variables, since we only need to contain the part of the unresolved dynamics that effects the resolved variables. In Section 2, the framework of the ORV method is described in details and the expected error with ORV derived. In Section 3, the complicated form of the modeling error equation and its gradient which will be used later for finding the best replacement variables R, are studied and simplified. Several ways for normalizing the ORV error and orthogonality constraints of the ORV system are also discussed. The techniques of Lagrange multipliers and unconstrained minimization used for minimizing the expected error of the ORV system are presented in Section 4. In Section 5, numerical results for random linear systems of ODEs and a scalar variable-coefficient PDE (a heat equation), are presented to illustrate the potential and issues in the ORV method. Discussion and conclusions are given in Section 6. We outline some directions and questions for future research in Section 7.

2 Setup and framework

In this work, we only consider the case of a linear system of ODEs. Nonlinear ODEs are a subject for future research. We shall make some simplifying assumptions:

1. a probability distribution $\rho(z)$ exists for z, which approximates the dynamics of the system (1.1) in the sense that for any set $S\in\mathrm{R}^{n+m}$, the likelihood of $z\in S$ is $\rho(S)$;
2. ρ is a reasonable approximation at all times $t>0$.

The probability distribution $\rho(z)$ is an important part of the framework, and is an input to the process-it is supplied by the user. It has to be selected using domain-specific knowledge and intuition, and will often be ad-hoc and imprecise. Therefore, it is a crucial requirement for the success of the optimal replacement variables method (ORV) that the method has to work well with such an imprecise, inaccurate probability distribution. This approximate or utility distribution is used only for computing the expectation of physical quantities one want to minimize. It is unlikely that samples of $z(t)$ obey this distribution, one only hopes that minimizing expected errors with respect to it, will result in algorithms with good results and/or improved performance.

Mathematically, ρ will be taken to be Gaussian distribution, that is,

$$z\sim\mathcal{N}(\mu,S), \tag{2.1}$$

with

$$S=\begin{pmatrix} S_0 & S_1 \\ S_2 & S_3 \end{pmatrix}. \tag{2.2}$$

It will be assumed to be zero-mean, i.e., $\mu=0$.

We shall construct a small set of variables to replace y, of the form

$$u=Ry, \qquad R\in\mathrm{R}^{k\times n}, \qquad k\ll n, \tag{2.3}$$

a reduced system vector

$$w=(x_*,\, u), \tag{2.4}$$

and a reduced ODE

$$w_t=F_*w, \qquad w(t=0)=\begin{bmatrix} x \\ Ry \end{bmatrix}. \tag{2.5}$$

Write

$$w_z=\begin{bmatrix} x \\ Ry \end{bmatrix}=Pz, \qquad P=\begin{pmatrix} I & 0 \\ 0 & R \end{pmatrix}, \tag{2.6a}$$

$$\frac{d}{dt}w_z = Pz_t = PFz, \tag{2.6b}$$

and now the goal is to pick R and F_* so that w mimics w_z as closely as possible.

For the measure of closeness, we pick the error e as

$$e = E\Big(\Big\|\frac{d}{dt}(w_z - w)\Big\|^2\Big) = \mathrm{E}\big(\|PFz - F_*Pz\|^2\big), \tag{2.7}$$

where the expectations are taken with respect to the Gaussian density on z (2.1), and where w is evaluated at $w = w_z$.

We call the variables u which minimize the closeness metric (2.7) *optimal replacement variables* (ORV) and we also use this as the name for the full system reduction method.

2.1 Expectation

The *expectation* of a random vector x having a probability density function $\rho(x)$ is defined as

$$\mathrm{E}(x) = \int x\rho(x)dx.$$

An important identity is that for any matrix M,

$$\mathrm{E}(Mx) = M\,\mathrm{E}(x). \tag{2.8}$$

Note that the analogous result is not generally true for nonlinear functions:

$$\mathrm{E}\big(f(Mx)\big) \neq M\,\mathrm{E}\big(f(x)\big). \tag{2.9}$$

But, if $f(x)$ is a low-degree polynomial, expressions for (2.9) exist and are not extremely complex. This is relevant to the extension of ORV to nonlinear systems of ODEs, a topic of future research.

The conditional expectation is defined similarly, except the conditional probability is used:

$$\mathrm{E}(x|y) = \int x\rho(x|y)dx.$$

The result (2.8) is true for conditional expectations as well.

It can be shown [5, Sec. 6.5] that for any function $v = v(x)$, the function $h(v)$ that minimizes

$$\epsilon(h) = \mathrm{E}\big(\|x - h(v(x))\|^2\big)$$

is

$$h_*(v) = \mathrm{E}\big(x|v\big) = \arg\min_h \big(\epsilon(h)\big).$$

2.2 Justification of assumptions

This work investigates the simplest possible case of the ORV method, namely a linear system of ODEs and a Gaussian distribution on the phase space. This is appropriate for a first presentation of the method, but we can justify these simplified assumptions a little more than that.

Nonlinear

While the most physically realistic and interesting systems of ODEs are nonlinear, many important systems are linear.

Fixed density

The justification of a time-invariant, zero-mean Gaussian distribution for the phase space density is somewhat lengthy and we defer it to Section 6.1.

Closeness metric

There are many possible versions of the closeness metric. Two of them are

$$e = E\big(\|x_t - E(x_t|w)\|^2\big), \qquad e = E\big(\|z_t - E(z_t|w)\|^2\big).$$

The first metric has the drawback of not requiring the replacement variables to reproduce their own dynamics with any accuracy, and this seems like it would not work-to approximate x_t well, only Ry needs to be accurate, but if the dynamics of y are all wrong, neither y or Ry will be accurate. The second metric goes too far in the other direction-it required that the dynamics of both x and y be preserved, even though most of the information in y is not needed. So the closeness measure (2.7) is a plausible compromise between these two extremes. Whether it is the best measure of closeness or what the best measure of closeness might be, seems to be an open question.

2.3 The expected error

We want to compute the best R,

$$R_* = \arg\min_R e(R). \tag{2.10}$$

We need to simplify the error (2.7) down to an expression that can be minimized numerically or analytically.

First, what is $E(z|w_z)$? The conditional of a zero-mean Gaussian (see [8]) is

$$\big(z|Pz\big) \sim \mathcal{N}(\mu', S'), \tag{2.11}$$

with

$$\mu' = E\big(z|Pz\big) = SP^T ZPz = Kw_z, \tag{2.12a}$$

$$S' = E\big(zz^T|Pz\big) - \mu'\mu'^T = S - SP^T ZPS, \tag{2.12b}$$

where

$$Z = (PSP^T)^{-1} \quad \text{and} \quad K = SP^T Z. \tag{2.13}$$

This gives

$$z - E\big(z|Pz\big) = \big(I - KP\big)z. \tag{2.14}$$

The reduced system of ODEs will be taken to be

$$\begin{aligned} w_t &= \mathrm{E}\big(PFz|w_z\big)\big|_{w_z=w} \\ &= PF\,\mathrm{E}\big(z|w_z\big)_{w_z=w} = PFKw, \qquad \text{giving} \quad F_* = PFK, \end{aligned} \tag{2.15}$$

so

$$\begin{aligned} e &= E\big(\|PF\big(z-E(z|Pz)\big)\|^2\big) = E\big(\|PF\big(I-KP\big)z\|^2\big) \\ &= \mathrm{tr}\big(PF(S-SP^TZPS)F^TP^T\big). \end{aligned} \tag{2.16}$$

3 The error equation

This section describes the unfolded and simplified equation for the expected error, and its gradient.

Consider three different parts of $e(R)$,

$$e(R) = e_0+e_1-e_2 = \mathrm{tr}(PFSF^TP^T)-\mathrm{tr}(PFSP^TZPSF^TP^T), \tag{3.1}$$

where we will accumulate in e_0, as we find them, all the terms from e_1 and e_2 that do not depend on R. Such terms can be ignored in the minimization. Note that for any partitioned matrix M:

$$\mathrm{tr}\begin{pmatrix} M_0 & M_1 \\ M_2 & M_3 \end{pmatrix} = \mathrm{tr}\big(M_0\big)+\mathrm{tr}\big(M_3\big).$$

Define

$$A = FSF^T = \begin{pmatrix} A_0 & A_1 \\ A_1^T & A_3 \end{pmatrix} \qquad \text{and} \qquad B = FS = \begin{pmatrix} B_0 & B_1 \\ B_2 & B_3 \end{pmatrix}. \tag{3.2}$$

Then, one has:

- e_0 gets a term $\mathrm{tr}(A_0)$,
- $e_1 = \mathrm{tr}(RA_3R^T)$,
- $e_2 = \mathrm{tr}(PBP^TZPB^TP^T)$.

By letting

$$H = BP^TZPB = \begin{pmatrix} H_0 & H_1 \\ H_1^T & H_3 \end{pmatrix}, \tag{3.3}$$

one has

$$e_2 = \mathrm{tr}(H_0)+\mathrm{tr}(RH_3R^T), \tag{3.4}$$

where

$$H_0 = B_0Z_0B_0^T+B_1R^TZ_1^TB_0^T+B_0Z_1RB_1^T+B_1R^TZ_3RB_1^T, \tag{3.5a}$$
$$H_3 = B_2Z_0B_2^T+B_3R^TZ_1^TB_2^T+B_2Z_1RB_3^T+B_3R^TZ_3RB_3^T. \tag{3.5b}$$

3.1 Unfolding Z

The inverse of a 2×2-partitioned symmetric matrix [9] such as S is

$$S^{-1}=\left(\begin{array}{cc} C_0+C_1C_3^{-1}C_1^T & -C_1C_3^{-1} \\ -C_3^{-1}C_1^T & C_3^{-1} \end{array}\right), \tag{3.6}$$

where the "half-inverse" matrix C is defined as

$$C=C(S)=\left(\begin{array}{cc} S_0^{-1} & S_0^{-1}S_1 \\ S_1^TS_0^{-1} & S_3-S_1^TS_0^{-1}S_1 \end{array}\right), \tag{3.7}$$

assuming that all the required submatrix inverses exist. If S is strictly positive definite this is guaranteed.

Defining $Q=(RC_3R^T)^{-1}$, gives

$$Z=\left(\begin{array}{cc} C_0+C_1R^TQRC_1^T & -C_1R^TQ \\ -QRC_1^T & Q \end{array}\right). \tag{3.8}$$

Define $X=R^TQR$, and then H_0 and H_3 can be unfolded as:

$$H_0=B_0C_0B_0^T+(B_0C_1-B_1)X(C_1^TB_0^T-B_1^T),$$
$$H_3=B_2C_0B_2^T+(B_2C_1-B_3)X(C_1^TB_2^T-B_3^T).$$

The term $B_0C_0B_0^T$ from H_0 is independent of R and will be moved into e_0 as

$$e_0=\text{tr}(A_0)-\text{tr}(B_2C_0B_0^T)=\text{tr}(F_1C_3F_1^T). \tag{3.9}$$

By defining

$$D_0=B_2C_0B_2^T, \qquad D_1=B_0C_1-B_1=-F_1C_3, \qquad D_2=B_2C_1-B_3=-F_3C_3, \tag{3.10}$$

the second error term e_2 can be expressed as

$$e_2=\text{tr}(RD_0R^T-D_1XD_1^T-RD_2XD_2^TR^T).$$

Taking the liberty of assuming that $Q=I$ (which will be discussed/justified later), one has

$$e_2=\text{tr}\big(R(D_0-D_1^TD_1)R^T-RD_2XD_2^TR^T\big),$$

and

$$E=A_3+D_0-D_1^TD_1=F_3C_3F_3^T-C_3F_1^TF_1C_3.$$

Note that E is symmetric.

The error function $e(R)$ becomes

$$e(R)=e_0+\text{tr}(RER_t)-\text{tr}(RD_2XD_2^TR^T). \tag{3.11}$$

It is important to note that if U is any $k\times k$ orthogonal matrix, then

$$e(R)=e(UR).$$

That is, the rows can be permuted, for example, without changing the error function which will be significant later.

The gradient

When minimizing $e(R)$, the gradient of e will usually be needed. Let R' be defined as in (A.2) in the Appendix, and then differentiate (3.11), giving

$$\begin{aligned} e' =& \mathrm{tr}\big(R'ER^T + RER'^T - R'D_2R^TRD_2^TR^T - RD_2R'^TRD^TR^T \\ & \qquad - RD_2R^TR'D_2^TR^T - RD_2R^TRD_2R'^T\big) \\ =& 2\mathrm{tr}\big(R'(ER^T - D_2R^TRD_2^TR^T - D_2^TR^TRD_2R^T)\big) \\ =& 2\big(ER^T - D_2R^TRD_2^TR^T - D_2^TR^TRD_2R^T\big). \end{aligned} \tag{3.12}$$

3.2 Expected errors and norms

The probabilistic framework can be used to calculate expected values of many quantities, such as:

- $E\|Fz\|^2 = E\|z_t\|^2 = \mathrm{tr}(FSF^T) = \mathrm{tr}(A_0) + \mathrm{tr}(A_3)$.
- $E\|x_t\|^2 = \mathrm{tr}(A_0)$.
- Error of replacing x_t by $E\big(x_t|x\big)$:

$$E\big\|x_t - E\big(x_t|x\big)\big\|^2 = \mathrm{tr}(F_1C_3F_1^T). \tag{3.13}$$

- $E\|Pz_t\|^2 = \mathrm{tr}(A_0) + \mathrm{tr}(RA_3R^T)$.
- The x-part of the ORV error: $E\|x_t - E(x_t|w)\|^2 = \mathrm{tr}(F_1C_3F_1^T) - \mathrm{tr}(RD_1^TD_1R^T)$.

Note that the last two properties depend on R.

These expressions are useful for normalizing the ORV error. It is more meaningful to compare the error of an ORV scheme to, for example the Galerkin approximation error or the expected value of w_t, than to simply look at the size of the numerical error obtained from the ORV method. This is particularly important when looking at the ORV error as parameters change.

This normalization is trickier than it might first look. The ORV-ed system will have errors in both x and Ry, so it is difficult to compare it with, say $E\|x_t\|$, which only involves x. This is especially a problem when looking at the behavior of the ORV system as the number of replacement variables changes. It is also difficult to compare the ORV error with $E\|z_t\|$ since the unresolved variables might have substantial dynamics that the ORV system does not (and does not need to) resolve.

Some plausible ways of normalizing the ORV error are

$$\frac{E\|x_t - E(x_t|w)\|}{E\|x_t\|}, \tag{3.14a}$$

$$\frac{E\|w_t - E(w_t|w)\|}{E\|Pz_t\|}. \tag{3.14b}$$

The first compares x errors only, which is not what ORV is minimizing although it is what we actually want to be small. The second compares w error from ORV with a quantity that depends on R in some random way, and which ORV has not tried to maximize. Both of these measures of ORV error are slightly problematic. But, we have not found a better way to normalized errors.

3.3 Orthogonality constraints

The dynamics of the replacement-variables system clearly depend only on the subspace spanned by R, not on R itself. Unfortunately, our error criterion $e(R)$ does depend on R. For example if R is multiplied by 2, then the u-part of the ORV error is also multiplied by 2. To deal with this in the minimization R needs to be normalized or constrained. An obvious way to do this is to require that

$$RR^T = I,$$

or alternatively, require

$$RGR^T = I,$$

for some symmetric positive definite matrix G.

If we allow ourselves to use a $G \neq I$, we might pick a convenient G. For example, the error term e_2 has R inside a matrix inverse, as well as terms of high degree in R. This greatly complicates the algebra, and will make any algebraic manipulation of $e(R)$ more difficult. This can be bypassed by changing the orthogonality condition from $RR^T = I$ to

$$RC_3R^T = I. \tag{3.15}$$

This gives $Q = I$ and $X = R^TR$. We will use (3.15) for the development of ORV method in the following discussion.

4 Minimizing the expected error

In this section, we will discuss numerical methods for minimizing the expected error of the ORV method.

4.1 Lagrange multipliers

The orthogonality condition (3.15) can be enforced using Lagrange multipliers with a constraint expression

$$l = \mathrm{tr}\big(L(RC_3R^T - I)\big), \tag{4.1}$$

where L is an $k \times k$ matrix.

Combining the error from (3.11) and the Lagrange multiplier term yields

$$e_l(R,L) = e_0 + \mathrm{tr}\big(RER^T - RD_2R^TRD_2^TR^T\big) + \mathrm{tr}\big(L(RC_3R^T - I)\big), \tag{4.2}$$

and the minimizer of $e(R)$ will satisfy the combined equation

$$0=\nabla_R e_l = ER^T + DR^T RD^T R^T + D^T R^T RDR^T + C_3 R^T L. \tag{4.3}$$

4.2 Minimization using the self-consistent field iteration

How can Eq. (4.3) be solved? It looks somewhat like a symmetric eigenvalue problem — Specifically, if

$$G(R)=E+DRR^TD^T+D^TRR^TD,$$

then (4.3) is

$$G(R)R^T-C_3R^TL=0,$$

which, except for the matrix depending on R, *is* a symmetric eigenvalue problem. This suggests the iteration

$$G(R_i)R_{i+1}^T=C_3R_{i+1}L_{i+1}, \tag{4.4}$$

which, if it converges, will converge to a minimizer of (4.2).

It turns out that minimization problems similar to (4.3) occur in electronic structure calculations, see [6]. The iteration (4.4) is used in that community, where it goes by the name of the *self-consistent field* algorithm (SCF). They find that it often either converges slowly or fails altogether to converge. Enhancements have been developed to make it more efficient and reliable, but minimization of this kind of expression remains an active area of research. It is used in the electronic structure community in spite of its poor performance because the problems are extremely high-dimensional and using methods involving higher derivatives is impractical.

We have tried the iteration (4.4) and like the electronic structure community, found that it often converges slowly or not at all. However it worked well enough to carry out a few experiments, which did give encouraging results.

4.3 Unconstrained minimization

The SCF iteration automatically handles the orthogonality constraints. A general minimization scheme such as Conjugate Gradient does not enforce constraints, although they could be introduced through the Lagrange multipliers.

Another way of dealing with the constraints is to create a new, lower-dimensional variable that incorporates the constraints automatically. This can be done in the following way: let

$$R=\begin{pmatrix} R_1 & R_2 \end{pmatrix}=V^{-1}\begin{pmatrix} I & Y \end{pmatrix},$$

where the I is a $k \times k$ identity matrix. Then the new variable Y is

$$Y = Y(R) = R_1^{-1} R_2.$$

R can be reconstructed from Y using the orthogonality constraint as follows:

$$\begin{aligned}
&RC_3R^T = I = V^{-1}\begin{pmatrix} I & Y \end{pmatrix} C_3 \begin{bmatrix} I \\ Y^T \end{bmatrix} V^{-T},\\
&M(Y) = VV^T = \begin{pmatrix} I & Y \end{pmatrix} C_3 \begin{bmatrix} I \\ Y^T \end{bmatrix},\\
&V = \operatorname{chol}\big(M(Y)\big),
\end{aligned}$$

where $\operatorname{chol}(M)$ is any matrix V satisfying $VV^T = M$. Hence,

$$R = R(Y) = \operatorname{chol}\big(M(Y)\big)^{-1}\begin{pmatrix} I & Y \end{pmatrix}, \tag{4.5a}$$

$$X = R^TR = \begin{bmatrix} I \\ Y^T \end{bmatrix} M^{-1} \begin{pmatrix} I & Y \end{pmatrix}. \tag{4.5b}$$

Note that the $R\big(Y(R)\big)$ will generally differ from R by an orthogonal transformation. This is desirable-the orthogonal transformation does not change the objective function.

How does this use of Y affect the gradient of $e(R)$? First, it can be shown that

$$\frac{\partial X}{\partial Y} = \operatorname{symm}\Big(A(Y)Y'B(Y)\Big), \tag{4.6}$$

where symm( ) is as defined in the appendix, and

$$A(Y) = \begin{bmatrix} I \\ Y^T \end{bmatrix} Q^{-1} \qquad \text{and} \qquad B(Y) = \begin{pmatrix} 0_{k\times(n-n)} & I_{n-k} \end{pmatrix}\big(I - C_3X\big). \tag{4.7}$$

Then it can be shown that

$$\frac{\partial e}{\partial Y} = 2B(Y)\Big(E - D_2XD_2^T - D_2^TXD_2\Big)A(Y). \tag{4.8}$$

The function $e(R(Y))$ and Eq. (4.8) can be used in a standard (unconstrained) minimization scheme such as conjugate gradient.

4.3.1 Pivoting

The matrix $M(Y)$ is not guaranteed to be well-conditioned. In experiments, it appears that even a small amount of ill-conditioning of $M(Y)$ can substantially slow down the convergence of conjugate gradient minimization.

How can the $R \leftrightarrow Y$ transformation be modified to improve its conditioning? Suppose

$$R = R(Y) = V^{-1}\begin{pmatrix} I & Y \end{pmatrix} Q, \tag{4.9}$$

where

$$Q=\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \qquad \text{and} \qquad W=Q^{-1}=\begin{pmatrix} W_1 & W_2 \end{pmatrix}, \tag{4.10}$$

are constant matrices to be described below. Then by defining

$$RQ^{-1}=RW=\begin{pmatrix} RW_1 & RW_2 \end{pmatrix}=(RW_1)\begin{pmatrix} I & (RW_1)^{-1}(RW_2) \end{pmatrix},$$

one has

$$Y=(RW_1)^{-1}(RW_2). \tag{4.11}$$

The inverse transformation is derived as follows:

$$I=RC_3R^T=V^{-1}\begin{pmatrix} I & Y \end{pmatrix}QCQ^T\begin{bmatrix} I \\ Y^T \end{bmatrix}V^{-T}, \tag{4.12a}$$

$$VV^T=M(Y)=\begin{pmatrix} I & Y \end{pmatrix}QCQ^T\begin{bmatrix} I \\ Y^T \end{bmatrix}, \tag{4.12b}$$

$$R(Y)=\text{chol}(M(Y))^{-1}\begin{pmatrix} I & Y \end{pmatrix}Q. \tag{4.12c}$$

As before, chol(M) is any matrix V satisfying $VV^T=M$.

Now matrices Q and W must be picked so that for a given R_*, cond$(R_*W_1)=1$. So

$$W_1=R^{\div}U, \tag{4.13}$$

for some orthogonal matrix U. This is the only requirement. U and W_2 can be chosen for convenience.

CG with pivoting

Pivoting would be added to a conjugate gradient minimization scheme as follows:

1. Choose an initial guess R_0.
2. Compute pivoting matrices W and Q based on R_0.
3. Do several iterations of CG minimization, giving an intermediate solution R_1.
4. Compute new pivoting matrices W and Q based on R_1.
5. If not yet converged, go back to Step 3 above.

In experiments, this substantially improved the convergence of the conjugate gradient minimization, and also seemed to improve the accuracy of finding the minimum.

5 Numerical results

In this section, we will present some preliminary results from applying ORV method to two test problems that illustrate the potential and issues in the implementation of the ORV method. All calculations present below were done using version 3.0.1 of Octave, a free Matlab-like program [10].

Random system of ODEs

The first test equation is the linear system of ODEs

$$z_t = Fz, \qquad z = (x,y), \tag{5.1}$$

with a random matrix F. Its entries were IID 0-1 Gaussian random numbers between zero and one. It was stabilized by computing its eigenvalues and adding the multiple of the identity that just makes the matrix stable.

The exact covariance matrix $\bar{S}$ for the problem was used: the instantaneous covariance matrix S was integrated forward in time by

$$S_t = FS + SF^T, \qquad S(0) = S_0 = I, \tag{5.2}$$

and then averaged:

$$\bar{S} = \frac{1}{T}\int_0^T S(t)dt. \tag{5.3}$$

A concern is the fact that in most cases we would not have nearly this much information about the phase space density, and so this test case might have anomalously good performance.

A way of testing this is to compare the performance of the Galerkin approximation and the $E(x_t|x)$ scheme, which differ precisely in their knowledge of the phase space density. In Fig. 2, we see that both schemes have about the same error-knowledge of the precise phase space density did not seem to help, and so the performance of the ORV scheme is probably realistic in this case.

Scalar variable-coefficient PDEs

The second test problem is a cosine expansion of the scalar variable-coefficient heat equation

$$u_t = \big(2+\cos(x)\big)u_{xx}, \qquad \text{with} \qquad u(x,t) = \sum_{k=0}^{\infty} u_k(t)\cos(kx). \tag{5.4}$$

This gives the tridiagonal system of ODEs for each coefficient $u_k(t)$

$$\frac{d}{dt}u_k = -\frac{1}{2}(k-1)^2 u_{k-1} - 2k^2 u_k - \frac{1}{2}(k+1)^2 u_{k+1}. \tag{5.5}$$

This is an infinite matrix, which is truncated at a fixed number $m+n$ of equations.

For the covariance matrix we used an exponentially-decaying diagonal matrix

$$S_{ij} = \alpha^i \delta_{ij}, \tag{5.6}$$

where the parameter $\alpha \in [0,1)$ is to be chosen to suit the experimenter's intuition. This is an ad-hoc choice, which is appropriate because often a more appropriate covariance matrix will not be known.

Note that in both test cases, the matrix R is computed once and used for all time. No attempt is made to follow the evolution of the covariance matrix in time in this study but this possibility will be investigated in the future work.

5.1 Performance results

It is not particularly meaningful to solve the ODEs $w_t = F_* w$, and $z_t = Fz$, look at the difference between the two solutions and say "look, they're small!" Small compared to what? To meaningfully evaluate the ORV system, its performance needs to be compared to other approximations of the original ODE.

Some reasonable approximations for comparison are Galerkin approximation, Random-R approximation, $m+k$ approximation, $\mathrm{E}(x_t|x)$ approximation.

• The Galerkin approximation is

$$x_t = F_0 x, \tag{5.7}$$

where F_0 is as defined in Eq. (1.1).

• Random-R approximation means picking a replacement matrix R at random and constructing a replacement-variable system using that R.

• The Expected-x_t approximation is

$$x_t = \mathrm{E}\big(x_t|x\big) = (F_0 + F_1 S_1^T S_0^{-1})x. \tag{5.8}$$

This uses the probability distribution ρ to improve the accuracy of the approximate dynamics over Galerkin.

• The $m+k$ approximation is for cases like the heat equation where there is an obvious reasonable way of defining the variables and an obvious ordering of them. In the case of a PDE with slowly-varying coefficients (such as our heat equation) the cosine modes of the solution are such variables. In this case a reasonable choice is to simply pick the k next modes in the sequence and let those be the additional degrees-of-freedom in the approximate dynamics.

Note that in most cases the $m+k$ approximation is not available.

5.2 Stability of the ORV system

The matrix F of the original system of ODEs will usually be a stable matrix. If the ORV matrix F_* is not also stable, then the ORV scheme will be of limited use. Fig. 1 tests this in the random-matrix test case. The values at the extreme right of the figure are for the original matrix F. The top curve is the spectral radius of F_*; $\rho(F_*)$, which is related to the degree of stiffness in F_*. Note that $\rho(F_*) < \rho(F)$ for all k, meaning that the ORV matrices are never any stiffer than the original matrix. The bottom curve is $\log(\rho(\exp(F_*)))$, which should be ≤ 0 for a stable matrix. It can be seen that F_* is stable for most values of k, but for $k = 5{,}6$ it is not quite stable. This tiny degree of instability would probably not have any negative effects unless time integration was carried out to quite large t.

Since this small instability is rare, if it were encountered, it could probably be eliminated by changing a parameter slightly (such as k) and recomputing R.

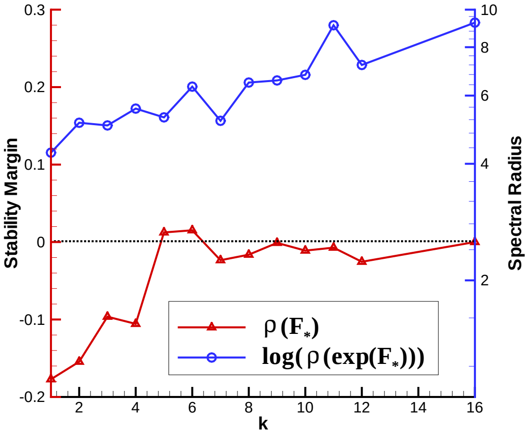

Figure 1: Spectral characteristics of F_* vs. k, $m=4$, $n=16$.

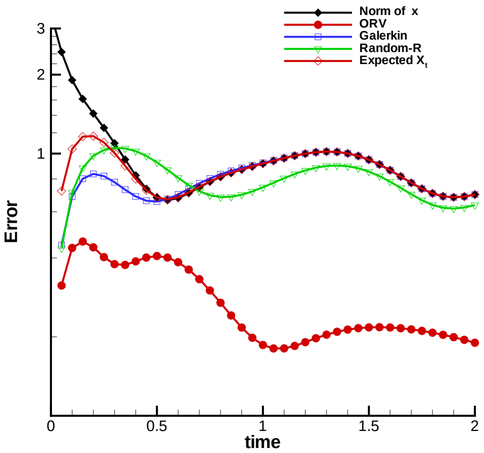

Figure 2: Errors in time for the random matrix problem with the ORV, Galerkin and random-R methods. $(m,n,k)=(8,40,12)$.

5.3 Accuracy of the ORV scheme

Fig. 2 compares the accuracy of the ORV scheme for the random matrix problem to the other three competitors-Galerkin, expected-x_t and the Random-R method. The ORV scheme is clearly superior to all of these. Note also that the accuracy does not degrade as t increases.

Fig. 3 shows the error of the ORV scheme for the random matrix problem, compared with the two different normalizations of the expected error from Eqs. (3.14a) and (3.14b). As k increases, the actual error decreases in a slow but exponential way. We also see that the numerical error and the two expected errors are all roughly similar.

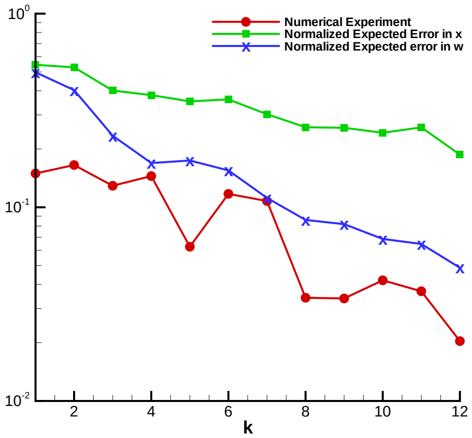

Figure 3: Expected and actual error vs. k for F_*.

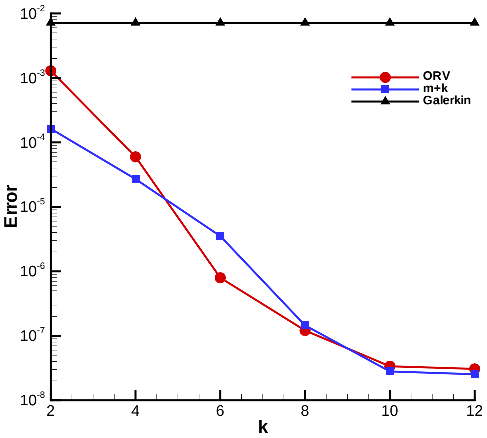

Figure 4: Accuracy of the ORV scheme versus number of replacement variables for the heat equation.

Fig. 4 shows the accuracy of the ORV version of the heat equation for several different values of k. The ORV system is compared the Galerkin approximation and also to the $m+k$ approximation.

The heat equation is a case where a reasonable set of additional variables exists. This makes the $m+k$ approximation a strong competitor to the ORV scheme. But in more general cases obvious replacement variables would not exist. The figure also shows that as k increases, the accuracy of the ORV scheme increases rapidly. The ORV scheme is, in fact, about the same accuracy as the $m+k$ approximation. It is better in one case, $k=6$. The ORV scheme is far better than the Galerkin scheme (no additional variables).

This can be viewed both as encouraging and discouraging. If we assume that the $k+m$ scheme is close to optimal, then this experiment shows that the ORV scheme has done as well as it is possible for it to do. But the discouraging viewpoint is that ORV was not superior to an obvious and simple competitor. We prefer the encouraging viewpoint.

5.4 Multiple starts and multiple non-global minima

The conjugate gradient minimization (CG) finds a local minimum, and if the function being minimized has multiple local minima, CG is not guaranteed to find the best one. Does our ORV error function have multiple minima? (Answer: sometimes.) And if so, what should be done about it? (Answer: That's an open question.)

The traditional, simple way of finding a global minimum amid (possibly) many local minima is with multiple random starts-make a random initial guess at the solution, use your favorite local minimizer starting there, save the result. Repeat this several times using different random initial guesses. If the solutions are all the same, you probably have a single global minimum. If some or all of them are different, you have multiple minima, and you use the best one.

We did this experiment on our ORV error function. We computed local minima of the ORV error function 200 times, counted the different solutions and looked at some statistics of each solution. The objective function value was normalized against the expected-x_t error of Eq. (3.13). The results are shown in Tables 1, 2 and 3.

First, the random ODE problem has only a few solutions, and the best one attracts almost the entire state space. One or two random starts would be enough to get the global minimum in this situation.

Table 1: $u_t=(2+\cos(x))u_{xx},\alpha_{init}=0.9,\alpha_{prob}=0.8$, 20 unresolved variables, 4 resolved, 4 replacement. Only some solutions are shown.

num	obj func	counts	cond(hess)	$\|RDR^t\|_F$	$\|R\|_f$	nn-dist
1	1.141×10^{-2}	4	1.403×10^{2}	1.211×10^{2}	3.93355	10.17232
2	5.417×10^{-2}	16	1.392×10^{2}	1.419×10^{2}	4.12778	16.10386
3	5.493×10^{-2}	4	1.441×10^{2}	1.676×10^{2}	4.33858	8.17578
4	5.493×10^{-2}	9	1.505×10^{2}	1.980×10^{2}	4.58933	8.06855
5	5.493×10^{-2}	10	1.618×10^{2}	2.329×10^{2}	4.88402	11.62701
6	5.493×10^{-2}	5	1.770×10^{2}	2.719×10^{2}	5.22763	8.06377
7	5.493×10^{-2}	5	1.964×10^{2}	3.150×10^{2}	5.62522	8.06386
8	5.493×10^{-2}	1	3.448×10^{2}	4.131×10^{2}	6.59003	46.52897
9	0.20361	7	1.420×10^{2}	1.540×10^{2}	4.26404	8.11335
10	0.20376	6	1.428×10^{2}	1.780×10^{2}	4.48230	24.97835
18	0.20751	8	1.618×10^{2}	2.516×10^{2}	5.15148	17.28774
26	0.20761	2	1.767×10^{2}	3.195×10^{2}	5.83747	16.09629
34	0.52970	3	1.771×10^{2}	2.823×10^{2}	5.42198	6.31713
42	0.53165	2	1.771×10^{2}	3.052×10^{2}	5.72282	6.42484
50	0.55186	1	2.888×10^{2}	3.377×10^{2}	6.02601	8.34724
58	0.87336	1	1.557×10^{2}	2.476×10^{2}	5.15070	7.70220
66	0.99997	1	6.007×10^{2}	5.019×10^{2}	8.09102	37.74330

Table 2: $u_t=(2+\cos(x))u_{xx}$,$\alpha_{init}=0.9$,$\alpha_{prob}=0.6$, 20 unresolved variables, 4 resolved, 4 replacement.

obj func	counts	cond(hess)	$\|RDR^t\|_F$	$\|R\|_f$	nn-dist
6.518×10^{-3}	37	1.337×10^{2}	1.211×10^{2}	10.00759	98.76203
3.310×10^{-2}	48	1.501×10^{2}	1.419×10^{2}	11.00082	1.692×10^{2}
3.634×10^{-2}	28	1.722×10^{2}	1.676×10^{2}	12.80773	59.50623
3.648×10^{-2}	5	2.160×10^{2}	1.980×10^{2}	15.25168	2.653×10^{2}
3.648×10^{-2}	3	4.758×10^{2}	2.329×10^{2}	18.46988	1.547×10^{2}
3.648×10^{-2}	1	1.364×10^{3}	2.719×10^{2}	22.62486	1.537×10^{2}
0.14753	21	1.722×10^{2}	1.780×10^{2}	13.36889	1.000×10^{11}
0.14753	12	1.504×10^{2}	1.541×10^{2}	12.26345	81.63501
0.14754	15	3.259×10^{2}	2.069×10^{2}	15.72647	45.25228
0.14755	5	8.935×10^{2}	2.404×10^{2}	18.86379	59.92150
0.14755	1	5.561×10^{3}	3.206×10^{2}	28.17489	1.033×10^{3}
0.16044	3	1.729×10^{2}	1.930×10^{2}	15.49814	35.34451
0.16045	4	3.860×10^{2}	2.197×10^{2}	16.55432	35.44029
0.16045	1	2.816×10^{3}	2.881×10^{2}	23.52398	1.051×10^{2}
0.16104	1	1.094×10^{3}	2.669×10^{2}	20.63488	4.268×10^{2}
0.16104	1	2.873×10^{3}	3.016×10^{2}	24.42192	6.444×10^{2}
0.45091	7	3.128×10^{2}	1.841×10^{2}	13.92223	44.82425
0.45092	1	7.447×10^{2}	2.122×10^{2}	16.19956	69.51190
0.45105	1	3.203×10^{2}	1.986×10^{2}	15.80370	1.596×10^{2}
0.45105	3	7.451×10^{2}	2.247×10^{2}	16.84083	96.04916
0.45105	2	1.896×10^{3}	2.559×10^{2}	19.80403	94.54004

Table 3: $u_t=Fx$,F random, 4 resolved variables, 16 unresolved, 4 replacement.

obj func	counts	cond(hess)	$\|RDR^t\|_F$	$\|R\|_f$	nn-dist
0.55702	182	1.064×10^{2}	6.35782	3.76880	1.000×10^{11}
0.55753	16	6.342×10^{2}	9.01687	4.81273	20.78350
1.56383	1	-0.93490	6.42993	3.62001	10.69016
1.74289	1	-1.79985	4.94075	3.46182	9.64476

The situation is completely different in the heat equation. Depending on parameter values, there can be a dozen or a hundred local minima, and the global minimum does not attract a very big section of the phase space. In this case, many random starts would be required to find the global minimum.

In the heat equation case, we observe that:

1. The best and good solutions occur at least somewhat more often than the bad solutions.

2. The best and good solutions are somewhat correlated with small values of the condition number of the Hessian matrix at the minimum, the norm $\|R\|_f$ and the quantity $\|RD_2R^T\|_f$. We do not know of any way of taking advantage of these facts, though, or why they are so.

3. There are many sets of several (completely different) solutions with the same objective function value. We also do not know why this is so.

6 Discussion and conclusions

This paper presents a method of system reduction, and shows on a couple of test cases that it is able to reduce the dimension of the system and still reproduce the dynamics of the original system with a useful degree of accuracy.

6.1 The fixed Gaussian assumption

This assumption is usually incorrect, of course. Even if the density was Gaussian at $t=0$, and the ODE is linear, the mean and variance of the density function usually will change with time.

For the density to be fixed we need

$$S_t = FS + SF^T = 0. \tag{6.1}$$

This can only be true if F has some pure imaginary eigenvalues-if it is neutrally stable.

We make the fixed Gaussian assumption for convenience, and because we want to study simple cases before considering more complex ones.

The real question is not whether the assumption is correct–it isn't. The real question is how much damage it does to the ORV scheme to use an incorrect phase-space density.

Our intuition is that in many cases the performance of the ORV method does not depend strongly on the details of the density, that it can be approximated quite crudely and the method will still work. The fact that our ORV scheme works reasonably well supports this conjecture.

This needs to be true, because in most cases only a crude approximation of the density would be possible. And if we are making crude approximations of the true density, we will make it a simple density, such as a time-invariant Gaussian, in the absence of additional information.

The following subsections discuss the reasoning behind this intuition.

6.1.1 Two kinds of probability

The two kinds are as follows: The first kind is the frequency that a certain event occurs, the second is a statement of one's knowledge about an event-it is how surprised one would be if the event occurred. When the weather forecast says "60% chance of rain," there is no random variable in play that determines whether it rains-it is a statement about the weather forecaster's knowledge of the weather. It means the forecaster would accept an even-money bet on the rain, but not 2-to-1 odds-he's not quite that sure.

In the case of the ORV method, the solution of the ODE is not in any sense a random variable. The probability is purely an expression of the user's knowledge of z.

6.1.2 Meaning of probability

In a crude sense, a probability distribution for something is a statement like "The object is probably in region A or B, but it might possibly be in region C, and I'd be very surprised to see it in region D."

A scheme for estimating something uses this information to decide how to hedge its bets-it knows to use most of its resources making the estimator accurate in regions A and B, less of its resources on region C, and very little effort on region D.

6.1.3 Result of errors in probability

Probability errors can be divided into *errors of ignorance* and *errors of misinformation*. Suppose a certain event will occur in region A or B, and suppose the forecaster has very little information about the event and can only say that it might be in A, B, C, D, or E. Then the estimator will be less accurate because it has to spread its resources over several extra regions. This is an error of ignorance.

Now consider the case where the forecaster has some incorrect information and thinks it will certainly be in either region A or C, but not B. Then when the event occurs in region B, the estimator will be completely wrong. This is an error of misinformation.

Errors of misinformation are generally more harmful than errors of ignorance.

6.1.4 Fixed Gaussian

Gaussian distributions are fairly wide and express little information. For example, there is a result [11] that among all probability distributions having a given covariance matrix, the Gaussian is the one with the largest entropy. Using a Gaussian distribution will probably result in mostly errors of ignorance and very few errors of misinformation.

As mentioned earlier, if the distribution used was fairly uninformative, the $E(x_t|x)$ scheme would have only slightly better performance than the Galerkin scheme, and in Fig. 2 we see that this is so.

So the use of a Gaussian density in our ORV method simply means that we are not willing (at this time) to put much effort into narrowing down what values the solution z is likely to take. The choice of a time-invariant Gaussian means we are not admitting to any knowledge of the dynamics of the ODE.

6.2 Numerical results

6.2.1 Accuracy

In one case (Random matrix) the ORV method is clearly superior to all of the competitors. In the other case (heat equation) a very strong competing method exists (the $m+k$ approximation) which ties the ORV method. All other competitors are soundly beaten. And in many cases the $m+k$ approximation is not available. For a new and novel scheme this seems like quite good performance, and potentially a strong competitor in the field of system reduction.

6.2.2 Stability

We see in Fig. 1 that the ORV method is usually stable, but occasionally it is slightly unstable. Note that the ORV method does not try to make the dynamics stable in any way, and the fact that it is usually stable is better than it might have been. Note also that Chorin's t-system often has severe stability problems.

One direction of extending ORV is to do a constrained minimization where the dynamic accuracy is optimized subject to the constraint that the system be stable. In most cases the constraint would not be activated at all. The remaining cases are only slightly unstable so we would expect the stability constraint to only cause a slight loss of accuracy.

In some cases, this slight risk of instability might be unacceptable. In a follow-on paper we will describe a variant of ORV using a different closeness criteria, and this version can have guaranteed stability with appropriate choice of parameters.

6.2.3 Long time accuracy

ORV achieves good accuracy for long times. This occurs even though the reduced dynamics are computed only once and do not track the changing solution in any way. This is superior to Chorin's t-system, which usually has good accuracy only for small time.

7 Further work

The work of this paper is preliminary and can be extended in many directions. These fall into several categories:

7.1 Orthogonality of R and variations on the error function

Does it matter to the performance of the ORV system what kind of orthogonality is used? If $k=1$ it makes no difference. Conjecture: if k is small, it makes only a small difference.

What about $RC_3R^T=\alpha I$? Then how would α be chosen?

Another direction for study is a matrix exponential version of the closeness metric.

7.2 Algorithms for minimizing the error function

Is $RC_3R^T=I$ really necessary? Can an effective CG minimization use some other constraint? It simplifies the algebra of the gradient greatly, but is it essential? How is speed of CG convergence affected by choice of orthogonality?

How can multiple local minima be dealt with? Is there a way to recognize and avoid them? Can the objective function be reformulated to have only a few, or only one minimum?

What about other minimization procedures, such as the Grassman CG algorithm proposed in [7]. Does this have any effect on the multiple local minimum issue?

7.3 Optimizing side conditions, such as sparseness of R and of F_*

The R that ORV computes is a full matrix. If the original F was very sparse, then F_* might have just as many non-zeroes as the original F did, in spite of being a smaller system of ODEs. How can be arrange for R or F_* to be a sparse matrix? (while still getting good system accuracy)

Another potentially-important question is how can one do the constrained minimization of requiring that F_* be stable or not too stiff, while also minimizing the error of the dynamics?

7.4 Scaling ORV up to large systems of ODEs

In this paper we have only considered systems of ODEs with small-to-moderate numbers of equations. To be most useful it needs to be usable on large systems. Does the ORV system give good performance in such cases? Is it computationally feasible to compute R? Are there issues that occur in large systems that do not occur in small ones?

7.5 Extending the ORV framework

There are several possible ways to extend the ORV framework, including:

1. Non-Gaussian probability distributions, such as mixtures of Gaussian. In many cases, a Gaussian probability distribution would depend on a few parameters, such as the decay α in our heat equation example. These are generally known at best approximately, and one idea is to let the probability distribution be a mixture of Gaussian with different parameters;

2. Nonlinear replacement variables. This might be done by augmenting the set of unresolved variables with a set of nonlinear functions of them, such as

$$r(y)=\{y_iy_j\}_{i,j=1,\cdots,n}, \tag{7.1}$$

and then computing replacement variables of the form

$$u=Ry+Nr(y).$$

Since the size of $r(y)$ grows rapidly with n, we might want to boil down the unresolved variables first with a 2-stage ORV process. That is, first compute $v=Ry$, and then compute $u=Nr(v)$ or $u=[v,Nr(v)]$.

7.6 Miscellaneous linear algebra

There are many questions that should be studied as well, such as

1. Do the ORV equations change or simplify in any interesting or useful way if F or S are diagonal or block-diagonal?

2. If they commute?
3. If a generalized singular value decomposition of F and S together is considered?
4. What if F or S is close to block-diagonal?

Acknowledgments

The authors dedicate this article to Prof. David Gottlieb. The area of this work is one that David felt interesting and important both theoretically and practically, before his passing.

The authors acknowledge the support of this work under the FRG grant FRG08-09-II-12 from the Hong Kong Baptist University and the RGC grant HKBU-200909 from Hong Kong Research Grant Council. The first author appreciates the hospitality of Prof. Wai-Sun Don and the Department of Mathematics at Hong Kong Baptist University during his visits.

Appendix

Properties of trace (tr) and symm( )

These are some identities that play an important role in the preceding results.

- $\mathrm{tr}(A)=\sum_i A_{ii}$. (definition)
- $\mathrm{tr}(AA^T)=\|A\|_F^2=\sum_{i,j}A_{ij}^2$.
- $\mathrm{tr}(AB)=\mathrm{tr}(BA)$; $\mathrm{tr}(A)=\mathrm{tr}(A^T)$.
- If $g(M)$ is a scalar function of a matrix M, and there is a $G(M)$ such that

$$\frac{\partial}{\partial s}g(M+sM')|_{s=0}=\mathrm{tr}\big(M'G^T(M)\big), \tag{A.1}$$

  then

$$\frac{dg}{dM_{ij}}=G(M)_{ij}. \tag{A.2}$$

- $\mathrm{symm}(A)=A+A^T$. (definition)
- $\mathrm{symm}(\alpha A+\beta B)=\alpha\mathrm{symm}(A)+\beta\mathrm{symm}(B)$.
- $A\ \mathrm{symm}(B)A^T=\mathrm{symm}(ABA^T)$.
- $\mathrm{symm}(A)=\mathrm{symm}(A^T)=\mathrm{symm}(A)^T$.

References

[1] A. J. Chorin and P. Stinis, Problem reduction, renormalization, and memory, Commun. Appl. Math. Comput. Sci., 1(1) (2005), 1–27.

[2] A. J. Chorin and O. H. Hald, Stochastic Tools in Mathematics and Science, Springer, 2006.

[3] A. Chertock, D. Gottlieb and A. Solomonoff, Modified optimal prediction and its application to a particle-method problem, J. Sci. Comput., 37 (2008), 189–201.

[4] D. Givon, R. Kupferman and A. Stuart, Extracting macroscopic dynamics: model problems and algorithms, Nonlinearity., 17(6) (2004), R55–R127, MR 2097022.

[5] J. F. Traub, G. W. Wasilkowski and H. Wozniakowski, Information-Based Complexity, Academic Press, 1988.

[6] C. Yang, J. C. Meza and L.-W. Wang, A trust region direct constrained minimization algorithm for the Kohn-Sham equation, SIAM J. Sci. Comput., 29(5) (2007), 1854–1875.

[7] A. Edelman, T. A. Arias and S. T. Smith, The geometry of algorithms with orthogonality constraints, SIAM J. Matrix. Anal. Appl., 20(2) (1998), 303–353.

[8] S. R. Searle, Linear Models, Wiley and Sons, 1997.

[9] T. Fortmann, A matrix inversion identity, IEEE Trans. Auto. Control., 15(5) (1970), 599–599.

[10] John W. Eaton, GNU Octave Manual, Network Theory Limited, 2002, isbn=0-9541617-2-6, also `www.octave.org`.

[11] T. M. Cover and J. A. Thomas, Determinant inequalities via information theory, SIAM J. Matrix. Anal. Appl. 9(3) (1988), 384–392.