First steps into Model Order Reduction

Alessandro Alla

impa

33º Colóquio Brasileiro de Matemática

First steps into Model Order Reduction

First steps into Model Order Reduction
Primeira impressão, julho de 2021

Publicado no Brasil / Published in Brazil.

ISBN 978-65-89124-33-7
MSC (2020) Primary: 62H25, Secondary: 65N06, 37M99, 37M05, 65D05

Coordenação Geral Carolina Araujo

Produção Books in Bytes **Capa** Izabella Freitas & Jack Salvador

Realização da Editora do IMPA
IMPA
Estrada Dona Castorina, 110
Jardim Botânico
22460-320 Rio de Janeiro RJ

www.impa.br
editora@impa.br

Contents

1 Finite Differences method **3**

1.1 Finite Differences for parabolic PDEs 6

1.2 Matlab code 7

2 The POD method **12**

2.1 Singular Values Decomposition 13

2.2 Proper Orthogonal Decomposition 15

2.2.1 Offline–online decomposition 16

2.2.2 Matlab code and comments 18

3 Discrete Empirical Interpolation Method **23**

3.1 Matlab code and comments 26

Bibliography **31**

Preface

Model order reduction (MOR) methods are of growing importance in scientific computing as they provide a principled approach to approximate many modern mathematical models of real-life processes, replace high-dimensional PDEs, with low-dimensional models. The dimensionality reduction provided by MOR helps to reduce the computational complexity and time needed to solve large-scale engineering systems enabling simulation based scientific studies not possible even a decade ago.

Examples of real-time simulation settings include control systems in electronics and visualization of model results while examples for a many-query (parameterized) setting can include optimization problems and design exploration. In order to be applicable to real-world problems, often the requirements of a reduced order model are:

- a small approximation error compared to the full order model,
- conservation of the properties and characteristics of the full order model,
- computationally efficient and robust reduced order modelling techniques.

Mathematically, MOR constructs low-dimensional subspaces, typically generated by the Singular Value Decomposition (SVD), where the evolution dynamics is projected. Thus, a high-dimensional system of differential equations is replaced by a low-rank model in a systematic fashion. Three steps are required for this low-rank approximation: (i) snapshots of the dynamical system for some time instances, (ii) dimensionality-reduction of this solution data typically produced with an SVD, and (iii) projection of the dynamics on the low-rank subspace. The first two steps are often called the *offline* stage of the MOR architecture whereas the third step is known as the *online* stage. Offline stages are exceptionally expensive, but enable the (cheap) online stage to potentially run in real time. This approach has been successfully applied to e.g. parametrized PDEs and optimal control problems.

A popular and well-established technique in MOR is Proper Orthogonal Decomposition (POD) which, in these notes, is introduced in a discrete setting. The notes can not replace a text book or research papers on the topic. Hopefully, they will be enough to get the reader excited and motivated to learn the topic more. Throughout the notes, we will discuss the method and its Matlab implementation. More information will be provided by the references[1] cited in the manuscript. At the end of the notes, we will list some possible applications of model order reduction methods.

The notes are structured as follows: In Chapter 1 we recall the finite difference method for a parabolic equation. In Chapter 2 we present the Proper Orthogonal Decomposition method and in Chapter 3 the Discrete Empirical Interpolation Method.

Acknowledgments. The author wishes to acknowledge IMPA for this opportunity, Carlos Tomei to support and revise this work. A deep gratitude to the colleagues who helped to discover, learn and appreciate this topic: Maurizio Falcone, Michael Hinze, J. Nathan Kutz and Stefan Volkwein.

[1] The list of references is by far not complete.

1 Finite Differences method

In this chapter we focus on the discretization of evolutive Partial Differential Equations (PDEs). We review some numerical schemes for PDEs, with emphasis on the finite difference method. We refer to the manuscripts Falcone and Ferretti (2013), Leveque (2002, 2007), and Quarteroni and Valli (1994), for finite differences, finite elements, semi-lagrangian and finite volume methods.

The semi-discretization of a PDE, say the spatial discretization, leads to a system of ordinary differential equations

$$\begin{aligned} M\dot{y}(t) &= Ay(t) + F(t, y(t)), \quad t \in (0, T], \\ y(0) &= y_0, \end{aligned} \tag{1.1}$$

where $y_0 \in \mathbb{R}^d$ is a given initial data, $M, A \in \mathbb{R}^{d\times d}$ given matrices and $F : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ a continuous function in both arguments and locally Lipschitz with respect to the second variable. It is well–known that under these assumptions there exists an unique solution for (1.1). Throughout these notes, we always assume that the model is given and known.

This wide class of problems arises in many applications, such as e.g. heat transfer or wave equations. In such cases, the dimension d is the number of grid points in the spatial discretization of the PDE and can be very large. The solution of system (1.1) may be computationally demanding and we will consider reduced order modeling techniques in the next chapters.

Let $y(t)$ be a smooth function of one variable. We approximate the time derivative $y_t(\hat{t})$ by a finite difference approximation based only on values of y in a neighbourhood of $\hat{t}$. For $\Delta t > 0$, the standard one sided approximations are given by

$$y_t(\hat{t}) \approx \frac{y(\hat{t} + \Delta t) - y(\hat{t})}{\Delta t}, \tag{1.2}$$

$$y_t(\hat{t}) \approx \frac{y(\hat{t}) - y(\hat{t} - \Delta t)}{\Delta t}. \tag{1.3}$$

Approximations (1.2) and (1.3) are of first order, whereas the following centered approximation

$$y_t(\hat{t}) \approx \frac{y(\hat{t} + \Delta t) - y(\hat{t} - \Delta t)}{2\Delta t}$$

is of order two. The verification uses the Taylor expansion of y at $\hat{t}$.

The centered approximation to the second derivative

$$y_{tt}(\hat{t}) \approx \frac{y(\hat{t} + \Delta t) - 2y(\hat{t}) + y(\hat{t} - \Delta t)}{\Delta t^2} \tag{1.4}$$

is also of order two.

Exercise. Compute approximations for the first and second derivative of $y(t) = e^t$ at $\hat{t} = 1$ for $\Delta t = \{0.1, 0.05, 0.025, 0.0125\}$. Verify the order of convergence.

The time discretization of (1.1) might be done in several ways, see Quarteroni, Sacco, and Saleri (2007). We begin by setting a temporal step size $\Delta t > 0$ and defining $t_k = k\Delta t \in [0, T]$, with $k = 0, \ldots, m$ and $t_m = T$. We will denote by $y(t_k)$ the continuous solution of (1.1) at time t_k, whereas by y^k the numerical approximation at time t_k. If the method converges $y_k \to y(t_k)$ when $\Delta t \to 0$.

To build a numerical scheme for (1.1) one might use formula (1.2), say a Taylor expansion around t_k for the time derivative and get the explicit Euler method:

$$M\frac{y^{k+1} - y^k}{\Delta t} = Ay^k + F(t_k, y^k), \quad y^0 = y_0, \qquad k = 0, \ldots, m-1. \tag{1.5}$$

This method is explicit: the unknown y^{k+1} only depends on the solution at the previous step y^k:

$$My^{k+1} = y^k + \Delta t(Ay^k + F(t_k, y^k)), \quad y^0 = y_0, \qquad k = 0, \ldots, m-1.$$

If M is not the identity matrix, this is a linear system at each iteration k.

The implicit Euler method is, on the contrary, built using a Taylor expansion around t_{k+1}. This leads to

$$M\frac{y^{k+1}-y^{k}}{\Delta t}=Ay^{k+1}+F(t_{k+1},y^{k+1}),\quad y^{0}=y_{0},\qquad k=0,\dots,m-1 \tag{1.6}$$

where it has been used (1.3) to discretize the time derivative.

The solution (1.6) is defined implicitly and requires the solution of a nonlinear equation. If we define the function

$$\mathcal{F}(x):=M(x-y^{k})-\Delta t\left(Ax+F(t_{k+1},x)\right), \tag{1.7}$$

our approximation problem at time t_{k+1} reads $\mathcal{F}(y^{k+1})=0$.

Due to the nonlinearity of the problem, we use Newton's method to compute y^{k+1}. Here, we recall the standard Newton's method, which makes use the computation of $J_{\mathcal{F}}(x)$ the full Jacobian of $\mathcal{F}(x)$. There is a large literature describing faster variants for inexact Newton's method (see e.g. Quarteroni, Sacco, and Saleri (ibid.)).

The Jacobian with respect to x is

$$J_{\mathcal{F}}(x):=M-\Delta t\left(A+J_{F}(t_{k+1},x)\right), \tag{1.8}$$

where J_F is the Jacobian of the nonlinear term F in (1.1).

Newton's method gives rise to the iteration below, with initial condition x_0,

$$J_{\mathcal{F}}(x_i)\delta_i=\mathcal{F}(x_i) \tag{1.9}$$

$$x_{i+1}=x_i-\delta_i. \tag{1.10}$$

We iterate until $\|x_{i+1}-x_i\|\leqslant\varepsilon$ for a prescribed tolerance ε. Each iteration requires the solution of a linear system of dimension $d\times d$. The choice of x_0 is crucial: it is well-known that the method converges quadratically if the initial condition is *close* to the solution, e.g. to compute y^{k+1} one might set the initial condition $x_0=y^k$. std]close to a solution

The explicit Euler method (1.5) and the implicit Euler method (1.6) have order of convergence equal to one. However, in the rest of the paper we will work with the implicit scheme which is more stable than the explicit method.

1.1 Finite Differences for parabolic PDEs

Let us now consider a one dimensional two points boundary value problem:

$$
\begin{aligned}
&\tilde{y}_t(x,t) = \alpha \tilde{y}_{xx}(x,t) + f(t,\tilde{y}(x,t)), \quad (x,t) \in (a,b)\times(0,T),\\
&\tilde{y}(x,0) = \tilde{y}_0(x)\\
&\tilde{y}(a,t) = 0 = \tilde{y}(b,t)
\end{aligned}
\tag{1.11}
$$

where $\tilde{y}(x,t) : [a,b]\times[0,T] \to \mathbb{R}$ is the unknown, satisfying zero-Dirichlet boundary conditions, $y_0(x) : [a,b] \to \mathbb{R}$ is the initial condition and $f : [0,T]\times\mathbb{R} \to \mathbb{R}$ is given.

Semi-discretization. Let us start with the spatial discretization of equation (1.11). We first choose a spatial step size $\Delta x > 0$ and set $x_i = a + (i-1)\Delta x$ for $i = 1,\ldots n$ and $x_n = b$. We denote by $y_i(t)$ the semi-discrete approximation of the continuous solution $\tilde{y}(x_i,t)$ at x_i with $y(t) : [0,T] \to \mathbb{R}^n$, and approximate the second derivative by the centered finite difference (1.4)

$$
\tilde{y}_{xx}(x_i,t) \approx \frac{y_{i-1}(t) - 2y_i(t) + y_{i+1}(t)}{\Delta x^2}, \qquad i = 2,\ldots,n-1. \tag{1.12}
$$

From the boundary conditions in (1.11), $y_1(t) = 0 = y_n(t)$. The semi-discretization in space of (1.11) leads to a system of ODEs as in equation (1.1), where

$$
A = \frac{1}{\Delta x^2}\begin{pmatrix}
-2 & 1 & & & & \\
1 & -2 & 1 & & & \\
 & 1 & -2 & 1 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & 1 & -2 & 1\\
 & & & & 1 & -2
\end{pmatrix}, \qquad
F(t,y(t)) = \begin{pmatrix}
f(t,y_2(t))\\
f(t,y_3(t))\\
\vdots\\
\vdots\\
f(t,y_{n-2}(t))\\
f(t,y_{n-1}(t))
\end{pmatrix},
$$

of dimension $d = n-2$, $A \in \mathbb{R}^{d\times d}$, $F(t,y(t)) \in \mathbb{R}^d$ and the matrix M is the identity matrix of dimension $d \times d$ in this context.

Exercise. How does the matrix A and the vector $F(t,y(t))$ look like in case of nonzero Dirichlet boundary conditions $\tilde{y}(a,t) = \beta$, $\tilde{y}(b,t) = \gamma$ and $\beta,\gamma \in \mathbb{R}$?

Let us now consider a two dimensional two points boundary value problem,

$$\begin{aligned}\widetilde{y}_t(\xi,t) &= \alpha\Delta\widetilde{y}(\xi,t) + f(t,\widetilde{y}(\xi,t)) \quad (\xi,t)\in\Omega\times[0,T],\\ \widetilde{y}(\xi,0) &= \widetilde{y}_0(x),\\ \widetilde{y}(\xi,t) = 0 &\qquad (\xi,t)\in\partial\Omega\times[0,T]\end{aligned} \tag{1.13}$$

where $\Omega\subset\mathbb{R}^2$ is an open set, $\xi=(\xi_1,\xi_2)\in\Omega$, $\widetilde{y}(\xi,t):\Omega\times[0,T]\to\mathbb{R}$ is the unknown, $y_0(\xi):\Omega\to\mathbb{R}$ is the initial condition and $f(t,\widetilde{y}(\xi,t)):[0,T]\times\mathbb{R}\to\mathbb{R}$ is a given function. The Laplace operator is $\Delta y(\xi,t) = y_{\xi_1\xi_1}+y_{\xi_2\xi_2}$. We discretize the derivatives in space following the one dimensional example. We use the same step $\Delta\xi>0$ for both ξ_1 and ξ_2, the notation $\xi_{ij}=((\xi_1)_i,(\xi_2)_j)$ and approximate $y\left(\xi_{ij},t_k\right)\approx y_{i,j}^k$. Then

$$\Delta y\left(\xi_{i,j},t_k\right)\approx\frac{y_{i-1,j}^k-2y_{i,j}^k+y_{i+1,j}^k}{\Delta\xi^2}+\frac{y_{i,j-1}^k-2y_{i,j}^k+y_{i,j+1}^k}{\Delta\xi^2}.$$

Using compact notations, we obtain the matrix $A\in\mathbb{R}^{n^2\times n^2}$

$$A=\frac{1}{\Delta\xi^2}\begin{pmatrix}T & I & & & & \\ I & T & I & & & \\ & I & T & I & & \\ & & \ddots & \ddots & \ddots & \\ & & & I & T & I\\ & & & & I & T\end{pmatrix},\quad T=\begin{pmatrix}-4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ & 1 & -4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -4 & 1\\ & & & & 1 & -4\end{pmatrix}, \tag{1.14}$$

with $I,T\in\mathbb{R}^{(n-2)\times(n-2)}$ and I is the identity matrix. Now, the dimension of the problem is $d=(n-2)^2$. The order of the matrix A follows the *natural row-wise ordering* std]natural row-wise ordering where we take the unknowns along the bottom row from left to right, $\{y_{11},y_{21},y_{31},\ldots,y_{n1}\}$ followed by the unknowns in the second row, $\{y_{12},y_{22},y_{32},\ldots,y_{n2}\}$, and so on.

1.2 Matlab code

In this section we provide the Matlab code for (1.13) with

$$\begin{aligned}\alpha &= 0.05,\\ \widetilde{y}_0(\xi) &= \sin(\pi\xi_1)\sin(\pi\xi_2),\\ f(t,y(t)) &= \mu(y(t)^2-y(t)^3),\\ \mu &= 10.\end{aligned} \tag{1.15}$$

We set $\Omega = [0,1]^2$, $T = 2$, $\Delta\xi = 0.0125$, $\Delta t = 0.05$. The solution at time $t = 0$ and $t = 2$ is given in the top of Figure 1.1, whereas the contour lines in the bottom of the same figure.

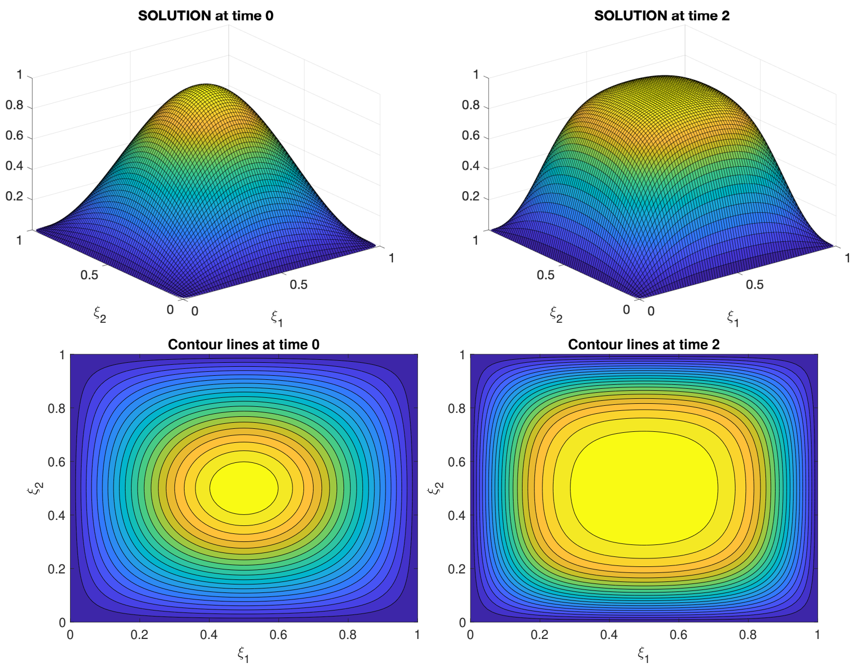

Figure 1.1: Top: Numerical approximation of (1.13) at time $t = 0$ (left) and $t = 2$ (right). Bottom: contour lines of (1.13) at time $t = 0$ (left) and $t = 2$ (right).

In the first part of the code we set the parameters used to define the problem.

```
clear
clc
close all

%Parameters
dx =0.0125;
PDE.mu = 10;
alpha = 0.05;
dt = 0.05;
```

```
x_tmp = 0:dx:1;
x = x_tmp(2:end-1);
y = x;
n = length(x);
t = 0:dt:2;
PDE.tol = 1e-5; %tolerance Newton's method

%Initial Condition
ic_cond_1 = sin(pi*x);
ic_cond = ic_cond_1'*ic_cond_1;
sol(:,1) = ic_cond(:);
```

Next, we discretize the Laplace operator. We note we used sparse matrices Matlab tools.

```
%Laplace discretization
e = ones(n,1);
A = spdiags([e -2*e e],-1:1,n,n);
A = kron(A,speye(n))+kron(speye(n),A);
PDE.A = alpha*A/dx/dx;
```

The functions $\mathcal{F}(x)$, $J_{\mathcal{F}}(x)$ in (1.7) and (1.8) are defined below. The Jacobian here is computed exactly, and defined as a sparse matrix due to the structure of the nonlinearity which is polynomial.

```
%Function and Jacobian for Newton's Method
full_sol =@(y,tmp,PDE)...
(tmp-y-dt*(PDE.A*tmp+PDE.mu*(tmp.^2-tmp.^3)));

df_full_sol = @(tmp,PDE)(speye(size(PDE.A))-...
dt*(PDE.A+spdiags(2*PDE.mu*tmp-3*PDE.mu*tmp.^2,0,n^2,n^2)));
```

Loop over time. We note that the initial condition in the Newton's method to compute y^{k+1} is the solution at the previous time y^k. That is true for $k = 0, \ldots, m-1$.

```
%Loop over time
tic
for i =k:length(t)-1
```

Títulos Publicados — 33º Colóquio Brasileiro de Matemática

Geometria Lipschitz das singularidades – *Lev Birbrair e Edvalter Sena*

Combinatória – *Fábio Botler, Maurício Collares, Taísa Martins, Walner Mendonça, Rob Morris e Guilherme Mota*

Códigos Geométricos – *Gilberto Brito de Almeida Filho e Saeed Tafazolian*

Topologia e geometria de 3-variedades – *André Salles de Carvalho e Rafał Marian Siejakowski*

Ciência de Dados: Algoritmos e Aplicações – *Luerbio Faria, Fabiano de Souza Oliveira, Paulo Eustáquio Duarte Pinto e Jayme Luiz Szwarcfiter*

Discovering Euclidean Phenomena in Poncelet Families – *Ronaldo A. Garcia e Dan S. Reznik*

Introdução à geometria e topologia dos sistemas dinâmicos em superfícies e além – *Víctor León e Bruno Scárdua*

Equações diferenciais e modelos epidemiológicos – *Marlon M. López-Flores, Dan Marchesin, Vítor Matos e Stephen Schecter*

Differential Equation Models in Epidemiology – *Marlon M. López-Flores, Dan Marchesin, Vítor Matos e Stephen Schecter*

A friendly invitation to Fourier analysis on polytopes – *Sinai Robins*

PI-álgebras: uma introdução à PI-teoria – *Rafael Bezerra dos Santos e Ana Cristina Vieira*

First steps into Model Order Reduction – *Alessandro Alla*

The Einstein Constraint Equations – *Rodrigo Avalos e Jorge H. Lira*

Dynamics of Circle Mappings – *Edson de Faria e Pablo Guarino*

Statistical model selection for stochastic systems – *Antonio Galves, Florencia Leonardi e Guilherme Ost*

Transfer Operators in Hyperbolic Dynamics – *Mark F. Demers, Niloofar Kiamari e Carlangelo Liverani*

A Course in Hodge Theory Periods of Algebraic Cycles – *Hossein Movasati e Roberto Villaflor Loyola*

A dynamical system approach for Lane–Emden type problems – *Liliane Maia, Gabrielle Nornberg e Filomena Pacella*

Visualizing Thurston's Geometries – *Tiago Novello, Vinícius da Silva e Luiz Velho*

Scaling Problems, Algorithms and Applications to Computer Science and Statistics – *Rafael Oliveira e Akshay Ramachandran*

An Introduction to Characteristic Classes – *Jean-Paul Brasselet* impa

Instituto de
Matemática
Pura e Aplicada

ISBN 978-65-89124-33-7

9 786589 124337