Spatiotemporal system identification with spectral methods

Khanmohamadi, Omid

2009

Khanmohamadi, O. (2009). Spatiotemporal system identification with spectral methods. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/18737>

<https://doi.org/10.32657/10356/18737>

SPATIOTEMPORAL SYSTEM IDENTIFICATION WITH SPECTRAL METHODS

OMID KHANMOHAMADI

SCHOOL OF MECHANICAL AND AEROSPACE ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR DEGREE OF

MASTER OF ENGINEERING

to E.A.T.

For your unconditional love, unfaltering support, and myriad sacrifices.

CONTENTS

Contents	ii
Acknowledgments.....	iv
Abstract.....	v
Outline	vi
CHAPTER 1. Introduction to System Identification	1
1.1 System Identification: Data-driven Modeling	1
1.2 Model Classes, Model Structures, Models, and all that	2
1.3 Parameter Estimation using Least Squares and QR Decomposition.....	8
1.3.1 Least Squares: from Pragmatic to Stochastic	10
1.4 System Reduction based on Error Reduction Ratio	14
CHAPTER 2. Spectral Methods in 1D, Periodic Domains	16
2.1 History of Spectral Methods	16
2.2 Finite Difference vs. Spectral Methods: Local vs. Global.....	17
2.3 Spectral Trial Functions and Test Functions	17
2.4 Periodic Domains and Fourier Spectral Methods	19
2.4.1 Continuous Infinite and Finite Fourier Expansions: Truncation Error.....	19
2.4.2 Differentiation of Continuous Expansion	20
2.4.3 Discrete Fourier Expansion: Even and Odd Quadratures	21
2.5 Interpolation Approach and Fourier Pairs	27
2.6 Periodic Sinc Function.....	29
2.7 Periodic Differentiation Tools: Matrices and Expansion Coefficients	30
2.7.1 Expansion Coefficients and Order of Wavenumbers: MATLAB Implementation.....	35
2.8 Continuous minus Discrete Fourier Expansion: Aliasing and How to Avoid it during Sampling	36
2.9 Approximation Error of Fourier Spectral Methods: Truncation plus Aliasing	39
CHAPTER 3. Spectral Methods in 1D, Non-periodic Domains	43
3.1 Non-periodic Domains and Polynomial Spectral Methods	43
3.2 Orthogonal Polynomials and Singular Sturm-Liouville Eigenproblems.....	44
3.2.1 Evaluating Chebyshev Polynomials using Three-Term Recurrence Relations.....	46

3.3	Differentiation of Continuous Chebyshev Expansion: Tridiagonal Recurrence Relations	48
3.4	Gauss Quadratures and Discrete Inner Products	49
3.5	Equivalence of Discrete Polynomial Expansion and Polynomial Interpolation	52
3.6	Discrete Chebyshev Expansion: Coefficients, Nodes, and Weights	53
3.7	Non-periodic Differentiation Tools: Matrices and Expansion Coefficients	55
3.8	Ameliorating Roundoff Effects in Polynomial Methods	60
3.9	Polynomial Interpolation: Lagrange, Newton, or Barycentric Form? That is the question.....	66
3.10	Lebesgue Function and Quality of Interpolants	71
3.11	Potential Function and Convergence rate of Interpolants.....	77
3.12	Approximation Error of Polynomial Spectral Methods: Truncation plus Aliasing Revisited	81
CHAPTER 4. Finite Difference Methods in 1D Domains		84
4.1	Rationale: Validation of Spectral Identification	84
4.2	Finite Difference Operators and Weights on Equispaced Grids	85
	4.2.1 Taylor Approximation.....	85
	4.2.2 Pade Approximation and a Symbolic Method for Calculating Weights.....	89
4.3	Finite Difference Weights on Arbitrarily Spaced Grids	94
CHAPTER 5. Identification of Spatiotemporal Differential Equations		97
5.1	Identification of Kuramoto-Sivashinsky Equation.....	99
5.2	Identification of Burgers Equation	108
5.3	Identification of Allen-Cahn Equation	110
Conclusion		118
References		121
Publications arising from this Work.....		125

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Daolin Xu, for his guidance and support throughout this work. My research has been supported by the Agency for Science Technology and Research (A*STAR) under an International Graduate Scholarship.

I am, forever, indebted to Prof. M.D. Greenberg whose one-of-a-kind Advanced Engineering Mathematics book opened a new vista ahead of me and whose teaching as well as words were decisive in my inclination towards the mathematical fundamentals of engineering. I would also like to thank Prof. L.N. Trefethen for giving me key hints for imposing boundary conditions efficiently. Prof. G. Strang's wonderful video lectures, papers, and books have been an eye-opener for me and I wish to thank him, again, for all he has done for the world of applied math.

I owe the little MATLAB I know to many people but I wish to explicitly thank E. G. Magrab, et al, for their "An Engineers' Guide to MATLAB," which built my foundation, Higham brothers for their "MATLAB Guide," which gave me a mathematician's view on MATLAB, L. Shure for her "Art of MATLAB," which has been teaching me a great deal about the innards of MATLAB, and P. J. Acklam for his "MATLAB array manipulation tips and tricks," the indispensable source to become a "MATLABician."

ABSTRACT

In inverse problem theory, the identification of nonlinear spatiotemporal systems is still an underdeveloped topic [1]. This work aims to introduce a means for *spatiotemporal system identification* based on *spectral methods*.

To achieve this goal, a continuous black-box model class is proposed and parameterized to obtain a model structure whose proper discretization yields a regression form, giving the unknown parameters based on the maximum likelihood estimation. An orthogonal system simplification algorithm is devised to eliminate the redundant parameters.

Spectral differentiation operators are extended to inverse problems for proper discretization of the continuous model structure. Trigonometric and algebraic spectral operators are introduced for periodic and non-periodic system identifications and for their practical implementation, wavenumber reordering and roundoff attenuation methods are proposed.

The main contribution of this work is to introduce a methodology, namely spectral spatiotemporal system identification, for one-dimensional periodic/non-periodic inverse problems and analyze its superior accuracy and robustness compared with its finite difference counterparts, through the numerical analysis of the method and experiments on indentifying Kuramoto-Sivashinsky (in chaotic regime), Burgers and Allen-Cahn spatiotemporal systems.

OUTLINE

System identification is an inverse approach to modeling, as opposed to the direct approach of modeling based on first principles, which is usually not feasible for the complex systems encountered in the real world: for biological, biomedical, environmental, transport, or economical systems, for example, we do not yet have the first principles that lead to reasonable models. Even when a model can be derived from first principles, it is necessary to adapt it to experiments; otherwise, because of simplifying assumptions in the derivation, there may be unacceptable discrepancy between the predictions arising from that model and the behavior of the true system. System identification, on the other hand, is a data-driven approach to modeling; it strives to extract a model from the behavioral data of a system. System identification for temporal processes (modeled with ordinary differential equations) has been studied extensively in literature but for spatiotemporal processes (modeled with partial differential equations) the topic is still rather underdeveloped. To the best of our knowledge, this work is the first to address the latter based on spectral methods.

Spectral methods are methods capable of approximating derivatives with exceptional accuracy. Derivative is a local property of a function and for its approximation it might seem sufficient to use only a few function values away from the point of interest. This argument, although reasonable for functions with slow variations, falls short of being true for functions with fast (spatial/temporal) variations as capturing those variations accurately by employing a very fine grid would require significant computational

resources. We have to resort to methods that are successful with coarser grids and spectral methods are such methods: they are *global*, meaning at any given point they use *all* the information available from the entire domain. Furthermore, for “smooth” enough functions, looking at the whole domain exploits the smoothness and results in “spectral” decay rates for the approximation error, which is faster than any polynomial decay rate. Contrariwise, local methods (such as finite differences) do not see past a certain number of points away from the point of interest and are therefore unable to see the smoothness, let alone exploit it.

Naturally, this manuscript includes chapters dealing with system identification and spectral methods. What is ostensibly unnatural is that a chapter is devoted to finite difference methods too. On the face of it, presenting finite difference methods might seem absurd as they were just deplored. However, finite difference methods will be used as a “test bed” for validating our spectral results.

We begin in CHAPTER 1 by rigorously defining a continuous model class and from its appropriate parameterization we show how to reach a continuous model structure. A special black box model class (based on function expansions) is then selected and customized in order to allow enough “complexity” for modeling spatiotemporal processes. Guided by the “information” from the model class and the observed data, and assuming the model structure has been discretized, we discuss how to “estimate” a model (i.e., estimate the parameters) for a particular process based on the least squares criterion for the “prediction” errors. We first treat least squares pragmatically (based on linear algebra only) and then stochastically (based on statistics + linear algebra) and show how the combination of these two viewpoints amounts to the *best linear unbiased estimates*

(BLUE) for observations with varying “reliability” and/or “relevance” due to, for instance, noise contamination and/or inapplicability of the model structure in some intervals. The model is at last made “parsimonious” by devising a system simplification algorithm, based on orthogonal decompositions, that eliminates the redundant parameters based on their insufficient contributions to the reduction of the prediction error.

CHAPTER 2 starts our journey through spectral methods in a single dimension by introducing *trigonometric (Fourier) spectral methods* suitable for system identification problems in *periodic domains*. After a bit of history and a brief comparison of finite difference methods with spectral methods, we embark on developing Fourier spectral differentiation operators by building on infinite, continuous, Fourier expansions and employing two approximations, namely truncation and discretization, to make that rather theoretical concept amenable to practice. We show that both these approximations are projections in effect and that the discretization involves even or odd trapezoidal quadratures to approximate the expansion coefficients, depending on whether there are an even or odd number of *equispaced* samples, respectively. It is then demonstrated that with the right trapezoidal quadrature employed for discretization, the discrete Fourier expansion is equivalent to interpolation in equispaced points with a periodic *sinc* basis and this fact is used to present matrix as well as recursion methods for implementing the Fourier spectral differentiation operator suitable for system identification. Finally the truncation and aliasing errors (due to the truncation and discretization approximations) are analyzed to indicate the “spectral” accuracy of Fourier spectral methods and a criterion is obtained to test for and avoid aliasing during sampling.

The discussion of spectral methods is continued (and in a sense complemented) in CHAPTER 3 with a shift from periodic to *non-periodic* domains. We demonstrate the need for a corresponding change of the expansion basis from trigonometric to *algebraic* and show how the family of *polynomials* that possesses the valuable property of orthogonality and the essential property of completeness can be obtained from singular Sturm-Liouville “eigenproblems.” Chebyshev polynomials are then chosen as a particular member of that family and are used as our algebraic basis. Again we show that the truncation of the infinite, continuous, Chebyshev expansion leaves us with a finite expansion to be discretized using quadratures; only this time instead of the equispaced trapezoidal quadrature we have to use Gauss quadratures based on *non-equispaced (clustered)* Chebyshev nodes and we indicate that the polynomial expansion discretized using these quadratures is equivalent to an interpolation with a Lagrange interpolation polynomial basis. Thereafter, the two viewpoints are utilized to obtain recursion and matrix expressions for the Chebyshev spectral differentiation operators. We point out the indispensability of sampling according to clustered nodes by resorting to Lebesgue functions for quantizing the quality of the polynomial interpolants in clustered and equispaced nodes and use potential theory to study the convergence rate of the corresponding interpolants to the true function. Having obtained the Chebyshev spectral operators in theory, we propose special implementation techniques for ameliorating roundoff artifacts, leading to efficient generation of the corresponding matrices on finite precision machines. At last, the approximation error of algebraic spectral methods is analyzed and a criterion is given to test against aliasing during clustered sampling.

CHAPTER 4 is devoted to finite difference methods but it is no traditional treatment of this topic for sure. We introduce five basic operators, namely forward difference, backward difference, center difference, shift, and averaging operators, acting on the elements of an equispaced complex sequence and using Taylor expansion we approximately represent the differentiation operator in terms of these *basic difference operators*. The relations between the basic operators themselves then allow us to express the resulting approximate differentiation operator in terms of just one basic operator, which is usually chosen to be the shift operator, and this way we deal with differential operators by approximating them in terms of difference operators with the advantage of having compact expressions for approximating high order differential operators to high orders of accuracy. On the other hand, sometimes it is more convenient to have expressions in term of the grid points themselves rather than ones involving operators acting on the grid points. To this end we introduce Pade approximation using which we come up with a concise *symbolic algorithm* for calculating finite difference weights that can approximate an explicit or implicit derivative equation on an *equispaced* grid which may or may not be “staggered.” As the last step we introduce a *numerical algorithm* that can handle *non-equispaced* grids too (needed as a validation tool for the results produced by algebraic spectral methods) with two clever recursion relations based on rearranging two equivalent representations of the Lagrange cardinal function.

CHAPTER 5 presents several numerical experiments which may be categorized based on the type of spectral methods involved (trigonometric or algebraic). We present three case studies, namely identification of Kuramoto-Sivashinsky, Burgers, and Allen-Cahn equations; they fully cover the breadth of the one dimensional methods introduced in this

thesis. The first two focus on system identification in periodic domains and the last deals with the non-periodic domains. In each of these case studies the parameters of the underlying system are estimated from only a time series of its simulated response and systematic studies of the estimate and prediction errors are done. Also analyzed are the errors in dominant coefficients and the sensitivity of the estimate errors to the time point choice and error reduction ratio threshold, the two parameters involved in state data selection and system simplification algorithms, respectively.

Finally we present a conclusion encapsulating the ideas proposed in this thesis for spatiotemporal system identification with spectral methods.

CHAPTER 1. INTRODUCTION TO SYSTEM IDENTIFICATION

1.1 SYSTEM IDENTIFICATION: DATA-DRIVEN MODELING

Obtaining a model capable of predicting a phenomenon has always been a problem of paramount importance in science. As it turns out, not always can we take a direct approach to that problem; constructing a model based on first principles and constitutive laws is usually *not* feasible for the complex systems encountered in the real world. We lack either the required principles/laws or the full set of conditions that should be imposed on them to yield the particular solution sought [2]. For instance, for biological/biomedical systems, environmental systems, or process control, we lack first principles that lead to reasonable models. Furthermore, even when we do *not* lack those principles or conditions and a model can be obtained based on them, there is always a risk of unreasonable discrepancy between the behavior predicted by the model and the true behavior of the system, resulting from too much approximation in deriving the model or the use of improper parameter values – the notorious simulation dilemma [3].

Consequently, we have to resort to an inverse approach instead of dealing with the modeling problem directly.

An inverse approach goes about finding the model of the system based on its input-output behavior. This is referred to as system identification. The work of Yule [4] on building an ad hoc linear model for the prediction of the sunspot cycle exemplifies one of the earliest uses of system identification. An even earlier, and probably more famous, example is the

successful prediction of the orbit of the asteroid Ceres in 1801 by Gauss which was based on the least squares.

At the heart of system identification lie three concepts: *information*, *estimation*, and *validation*, all of which deal with the notion of “model.” This clearly calls for a rigorous definition of model.

1.2 MODEL CLASSES, MODEL STRUCTURES, MODELS, AND ALL THAT

A *model* describes the relation between observations and allows for predictions. It is typically denoted by m . Rigorously, suppose that Z^t are the input (u)-output (y) data available up to and including time t then a model m of the system is a sequence of functions $g_m(t, Z^{t-1})$ which predict the value of the output y at time t based on the past input-output measurements

$$m : \hat{y}(t | t-1) = g_m(t, Z^{t-1}). \quad (1.1)$$

A model that only contains the predictor function is called a *predictor model*. Clearly, we must allow for the possibility that the model may *not* be able to completely predict the output $y(t)$ from past data; some *prediction error* will be present, equal to the part of the output that the model cannot describe:

$$\varepsilon(t) = y(t) - \hat{y}(t | t-1). \quad (1.2)$$

If, in addition to the predictor function, some measure of the prediction errors is incorporated into the model, we will then have a *probabilistic model*. A probabilistic model can be either complete, when the predictor model is complemented with the conditional PDF of the prediction errors $f_\varepsilon(x, t, Z^{t-1})$, or partial, when instead of the

complete PDF only its second moment (variance $\lambda(t)$ in one dimension or covariance matrix $\Lambda(t)$ in multiple dimensions) is incorporated into the predictor model.

Eq. (1.1) defines *one* model for the system in question. The goal of system identification is to identify the “best” model and one way to find it is to look for it in a *model class* $\mathfrak{M}^*$ (also known as a *model class*) – a confined collection of “related” models.

Normally a model class is uncountable and, to perform a search for the best model, a way of “indexing” the members of the model class is required. The idea is to parameterize (index) the set “smoothly” over a “nice” area and do the search over the parameter set (index set) [5]. The parameterized model class $\mathfrak{M}^*$ is the *model structure* $\mathfrak{M}$. The indices are usually denoted by the vector θ . These statements are squeezed in

$$\mathfrak{M}: \theta \rightarrow g(t, Z^{t-1}; \theta) \in \mathfrak{M}^* \quad \theta \in D_{\mathfrak{M}} \subset \mathbb{R}^d. \quad (1.3)$$

where d is the dimension of the parameter vector. In other words, a model structure $\mathfrak{M}$ is a mapping from the parameter space θ to a model class $\mathfrak{M}^*$. Now $\mathfrak{M}(\theta)$ would denote the particular model associated with θ ,

$$\mathfrak{M}(\theta): \hat{y}(t|t-1; \theta) = g_m(t, Z^{t-1}; \theta). \quad (1.4)$$

With every model class is associated a complexity, C , which is a measure of how “flexible” the set is. A proper model class is flexible enough to blanket all kinds of reasonable system behavior and uses “physical insight” to keep the flexibility from running amok.

If the dependence of g on θ is linear and the subset $D_{\mathfrak{M}}$ is the whole $\mathbb{R}^d$, then the associated model structure is dubbed a *linear regression*; that is,

$$g(t, Z^{t-1}; \theta) = \varphi^T(t, Z^{t-1})\theta + \mu(t, Z^{t-1}) \quad (1.5)$$

is a linear regression model structure.

As an abstraction, it is convenient to assume a *true description*, S , for the system, with the same intrinsic properties as the model but of more complexity. The true description provides a means for assessing the quality of the associated model.

Real world systems are nonlinear and model classes for them constitute a palette of gray shades that run from “white” to “black” [6].

- *White (Box) Models* come from meticulous modeling based on first principles. This approach is also known as *physical modeling* and is the way object oriented modeling software like MODELICA and SIMULINK go about the modeling task.
- *Off-white Models* are those “white models” with some unknown parameters. In general, that is for nonlinear identification problems, approximating these parameters from observations is not easy.
- *Smoke Gray Models* are based on *semi-physical modeling*: basic physical understanding of the phenomenon being modeled is exploited to incorporate the probable nonlinearities in the model. In other words, nonlinear transformations are found that transform the observations in such a way that the results can describe the system in a linear fashion [5].
- *Steel Gray Models* encompass *composite local models* and *linear parameter varying* (LPV) models.

Composite linear models work by linearization around working points. Usually a nonlinear system has more than one working point of interest and a global model for it can be created from local linear models, each being a good approximation in the neighborhood of the corresponding working point. That is why these models

are also known as local linear models. If partitioning into neighborhoods is done according to a *regime variable* $\rho(t)$ with $\rho_k, k=1, \dots, d$ and the weight for the ρ_k -th neighborhood is denoted by $w_k(\rho)$ then the predicted output will be given by the sum of the local linear model each contributing to the sum according to its corresponding weight

$$\hat{y}(t; \theta) = \sum_{k=1}^d w_k(\rho) \hat{y}^{(k)}(t; \theta^{(k)}). \quad (1.6)$$

Linear parameter varying models [7], [8] are related to local linear models and their state space representation is

$$\begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + B(\rho(t))u(t) \\ y(t) &= C(\rho(t))x(t) + D(\rho(t))u(t) \end{aligned} \quad (1.7)$$

- *Slate-Gray Models* comprise of *hybrid models* and *block-oriented models*.

Hybrid models switch between different modes as the state varies. For them, the exogenous (regime) variable becomes a known function of state. If the partitioning is known, the problem simplifies to linear regression; otherwise mixed integer and linear/quadratic programming are required [9].

Block-oriented models are built up from fundamental blocks which are either linear dynamic systems or nonlinear static systems. The way these fundamental blocks are put together decides the final system. Two famous combinations are the Hammerstein model with the nonlinear static block as an actuator and the Wiener model with the nonlinear static block as the sensor.

- *Black (Box) Models* are based on function expansions. An expansion

$$\sum_{k=1}^m \theta_k g_k(\varphi) \quad (1.8)$$

is chosen in such a way that it can well estimate any function $g_0(x)$ relevant to the phenomenon under study.

We obtain Taylor expansion (known as Volterra expansion [10] in modeling jargon) with

$$g_k(\varphi) = \varphi^k. \quad (1.9)$$

The “best” basis functions $g_k(\varphi)$, however, are generated surprisingly from dilations, β_k , and translations, γ_k , of a mother function $\kappa(\varphi)$

$$g_k(\varphi) = g_k(\varphi, \beta_k, \gamma_k) = \kappa(\beta_k(\varphi - \gamma_k)). \quad (1.10)$$

This is as flexible as it can get and, based on the choice of the mother function, dilation parameters and translation parameters, it can yield Fourier expansion, wavelets, radial basis neural networks, least square support vector machines, etc [5]. Because the mother basis function is repeated a lot of times in the expansions used in black box models, these model classes are also referred to as *networks*.

In single variable treatments, functions like Volterra and Fourier basis belong to a *global* class as they have extensive variations over the whole domain in question.

On the other hand, expansion bases such as Gaussian bell $\kappa(x) = \frac{1}{2\pi} e^{-x^2/2}$ and

sigmoid $\kappa(x) = \frac{1}{1 + e^{-x}}$ are categorized as *local* as their significant variation

occurs locally.

One way to construct basis functions in multiple dimensions is to use *tensor products* of the appropriate single variable basis functions. There are at least two other common methods that are based on single variable basis functions: *radial construction* $\kappa(\|\varphi - \gamma_k\|_{\beta_k})$ and *ridge construction* $\kappa(\beta_k^T \varphi + \gamma_k)$. Clearly, the basis functions resulting from the former construction depend only on the distance of the state vector from a corresponding center state and those constructed by the latter depend only on the distance of the state vector from a corresponding hyperplane.

We will be dealing with phenomena that can be described by systems of partial differential equations (PDEs).

A one dimensional PDE may be represented by

$$\frac{\partial^m u(t, x)}{\partial t^m} = \sum_{j=1}^C \left(\theta_j x^{p_{1,j}} u^{p_{2,j}} \left(\frac{\partial u}{\partial x} \right)^{p_{3,j}} \left(\frac{\partial^2 u}{\partial x^2} \right)^{p_{4,j}} , \dots , \left(\frac{\partial^n u}{\partial x^n} \right)^{p_{n+2,j}} \right) \quad (1.11)$$

which is of order m in time derivative and n in spatial derivatives. The RHS of Eq. (1.11) is formulated by taking all linear combinations of the terms formed by taking every possible product of powers of the spatial variable, x , dynamical variable, u , and spatial derivatives of dynamical variable; the powers, $p_{i,j}$, are nonnegative integers and for each term they add to at most the degree of nonlinearity, $DoNL$, we assume for the system. The weights θ_j of this linear combination (i.e., the coefficients of the Volterra expansion) are the parameters to be estimated. Once we choose the highest order of spatial derivatives and the degree of nonlinearity of the system, the number of terms in Eq. (1.11) is fixed

and equals $C = (DoNL + n + 2)! / (DoNL!(n + 2)!)$, which is $C_{n+2}^{DoNL+n+2}$, the number of distinct combinations of variables up to $DoNL$.

Comparing the form of Eq. (1.9) and Eq. (1.11), it is seen that a Volterra expansion is a suitable model class for our problem. The resulting model structure will be a linear regression. If Y denotes a vector that contains the time derivatives and Φ is a matrix

whose j -th column is $x^{p_{1,j}} u^{p_{2,j}} \left(\frac{\partial u}{\partial x} \right)^{p_{3,j}} \left(\frac{\partial^2 u}{\partial x^2} \right)^{p_{4,j}} \dots \left(\frac{\partial^n u}{\partial x^n} \right)^{p_{n+2,j}}$, then

$$\hat{Y} = \Phi \hat{\theta} \quad (1.12)$$

is the linear regression predictor model for Y . Each row of $\hat{Y}$ and Φ correspond, respectively, to $\hat{y}(t; \theta)$ and $\varphi(t)^T \hat{\theta}$ for a point in time. Clearly, each column of Φ corresponds to a point in space. Using the material in CHAPTER 2, CHAPTER 3, and CHAPTER 4, we will explain in CHAPTER 5 how the elements of the vector $\hat{Y}$ and matrix Φ are approximated from samples of the function u

Now that we have the model the next step is to solve it for the parameters.

1.3 PARAMETER ESTIMATION USING LEAST SQUARES AND QR DECOMPOSITION

In section 1.2, we mentioned in passing that the goal of system identification is to identify the “best” model for a system. The quest for the best surely has to pass through the “goods” and a criterion has to be defined to compare different models in goodness. We have stated, repeatedly, that the essence of a model is its prediction ability so it is reasonable to postulate that a good model must be good at predicting. We mentioned that unless the model we have is the true description of the system, the prediction given by it will not be perfect and some prediction error is involved, $\varepsilon(t) = y(t) - \hat{y}(t | t-1)$. It is

logical then to require that this error be “small” for the model to be good. The question is now how to define smallness of the prediction error as it is not a scalar; at each point in time, prediction error is a vector, of the same size as that of the outputs. We choose to assign a “magnitude” to this vector by using a (suitable) norm and then summing those magnitudes over time to get the total error incurred. We use the norm

$$V(\theta, Z) = \sum_t \ell(\varepsilon(t, \theta)), \quad (1.13)$$

where $\ell(\cdot)$ is a scalar valued function. Thus the best parameter estimate leads to a model which minimizes that norm,

$$\hat{\theta} = \arg \min_{\theta \in \mathcal{D}_{\Theta}} V(\theta, Z). \quad (1.14)$$

Typically a quadratic norm is chosen as a measure of magnitude, in which case $\ell(\cdot)$ becomes

$$\ell(\varepsilon(\theta)) = \frac{1}{2} \varepsilon(\theta)^T \varepsilon(\theta) \quad (1.15)$$

Substituting this in Eq. (1.13), we obtain,

$$V(\theta, Z) = \|Y - \Phi\theta\|_{\ell^2}^2 = (Y - \Phi\theta)^T (Y - \Phi\theta) \quad (1.16)$$

and Eq.(1.14) becomes a least squares problem, seeking the minimizing parameter in Eq. (1.16). The “winner” is given by the *normal equations*

$$\Phi^T \Phi \hat{\theta} = \Phi^T Y. \quad (1.17)$$

Normal equations are always more badly conditioned than the original system of (over-determined) equations they are invoked to solve since the condition number of $\Phi^T \Phi$ is square of that of Φ . Hence, we use an alternative approach and compute the parameter estimates by introducing a *QR* decomposition [11] for the Φ ,

$$\Phi = QR. \quad (1.18)$$

in which, in reduced form, R is a $d \times d$ upper triangular matrix and Q has orthonormal columns; that is $Q^T Q = I$.

Solving Eq. (1.17) now becomes equivalent to solving the system

$$Q\beta = Y, \quad R\hat{\theta} = \beta. \quad (1.19)$$

The triangular matrix R makes the solution of the second part of Eq. (1.19) simply a back substitution.

It is noted that Q is not used in computing the parameter estimate and that using Householder reflections or Givens rotations, we are able to solve the least squares problem using just the idea of QR , without actually computing Q ; this is in contrast to using Gram-Schmidt that explicitly computes Q [12].

1.3.1 LEAST SQUARES: FROM PRAGMATIC TO STOCHASTIC

We have not used any statistical arguments in our prediction error minimization framework. Everything was presented in a pragmatic manner and in fact the power of this framework lies in its making sense without a probabilistic setting; a probabilistic setting would require that the experiments be repeated infinitely under similar conditions which would not be feasible as only a given set of data is available to us.

The independence of a pragmatic approach to least squares from an approach based on probability theory was realized as early as 1809, by Gauss [13], "...the principle that the sum of the squares of the differences between the observed and computed quantities must be minimum may be considered independently of the calculus of probabilities." Notwithstanding this, it is instructive to brief the fundamentals of the statistics for least

squares parameter estimation, as they shed light on such important areas as validation and experiment design.

Before embarking on our introduction of the aforementioned statistics, let us introduce a mild generalization into the least squares problem. Suppose observations are of either varying *reliability* or *relevance*. Then we have to weigh them differently. A neat way is to attach weights to prediction errors and a weight matrix W multiplied by the prediction error vector will do the trick:

$$V(\theta, Z) = \|W(Y - \Phi\theta)\|_{\ell^2}^2 = \|(WY - W\Phi\theta)\|_{\ell^2}^2. \quad (1.20)$$

It is seen in this new least squares criterion that both the output and the regression matrix are weighed by the weight matrix and hence the normal equation, giving the winner, changes accordingly to

$$(W\Phi)^T (W\Phi) \hat{\theta} = (W\Phi)^T WY, \quad (1.21)$$

or in explicit form,

$$\hat{\theta} = ((W\Phi)^T (W\Phi))^{-1} (W\Phi)^T WY. \quad (1.22)$$

Let us now introduce a disturbance B into the picture. Suppose that B takes care of the random part of the observations Y and that it has the statistical properties

$$\mu(B) = E[B] = 0 \quad \text{zero mean,} \quad (1.23)$$

and

$$V = \text{cov } B = E[BB^T] \quad \text{disturbance covariance matrix.} \quad (1.24)$$

Observations can now be written as the sum of a deterministic part $G(\Phi)$ and a stochastic part B ,

$$Y = G(\Phi) + B. \quad (1.25)$$

Notice that Eq. (1.25) is to describe the observations and not the predictions. That is why, despite our model structure being a linear regression, we have chosen to use the general $G(\Phi)$ instead of the linear $\Phi\theta_0$ (θ_0 is the vector of true parameters). We keep with the general function $G(\Phi)$ as long as it does not cause too much agony in our treatments.

If we substitute Eq. (1.25) (with a linear regression function just for now) in Eq. (1.22), we obtain

$$\begin{aligned}\hat{\theta} &= \left((W\Phi)^T (W\Phi) \right)^{-1} (W\Phi)^T W (\Phi\theta_0 + B), \\ &= \theta_0 + \overbrace{\left((W\Phi)^T (W\Phi) \right)^{-1} (W\Phi)^T WB}^{\tilde{\theta}}, \\ &= \theta_0 + \tilde{\theta},\end{aligned}\tag{1.26}$$

where $\tilde{\theta}$ is called the *parameter error*. If $(W\Phi)^T (W\Phi)$ is not singular and $(W\Phi)^T WB$ is zero, then the parameter estimate will approach the true parameter as the number of observations increases and that is *consistency*. The former requirement translates to $W\Phi$ having full rank and the latter to the “independence” between Φ and B . The former is a deterministic requirement as is the matrix $W\Phi$ whereas the latter is a stochastic condition just like the vector $(W\Phi)^T WB$ that imposes it.

We may just as readily derive the condition for the parameter estimate to be unbiased by calculating the expectation of Eq. (1.22)

$$\begin{aligned}E\hat{\theta} &= \overbrace{\left((W\Phi)^T (W\Phi) \right)^{-1}}^{\text{deterministic}} (W\Phi)^T W E \left(\overbrace{\begin{pmatrix} \hat{G}(\Phi) \\ \tilde{Y} \end{pmatrix}}^{\text{deterministic } \hat{G}(\Phi) + B} \right) \\ &= \left((W\Phi)^T (W\Phi) \right)^{-1} (W\Phi)^T W G(\Phi).\end{aligned}\tag{1.27}$$

If a linear structure can describe the observations (that is if $G(\Phi) = \Phi\theta_0$), Eq. (1.27) simplifies to $E\hat{\theta} = \theta_0$; that is, if a true description of linear regression type is available, the parameter estimate will be *unbiased* [14]. This leads to a new insight into the parameter error:

$$\begin{aligned}\tilde{\theta} &= \hat{\theta} - \theta_0, \\ &= \hat{\theta} - E[\hat{\theta}], \\ &= \left((W\Phi)^T (W\Phi) \right)^{-1} (W\Phi)^T WB.\end{aligned}\tag{1.28}$$

Eq. (1.28) yields the *parameter estimate covariance matrix*

$$\begin{aligned}P &= \text{cov } \hat{\theta} = E[\tilde{\theta}\tilde{\theta}^T], \\ &= \left((W\Phi)^T (W\Phi) \right)^{-1} (W\Phi)^T \overbrace{W E[BB^T]}^V W^T (W\Phi) \left((W\Phi)^T (W\Phi) \right)^{-1}.\end{aligned}\tag{1.29}$$

If all observations have the same reliability and relevance ($W = I$) this simplifies to

$$\begin{aligned}P &= \Phi^{-1} V (\Phi^T)^{-1}, \\ &= (\Phi^T V^{-1} \Phi)^{-1}.\end{aligned}\tag{1.30}$$

Eq. (1.30) is very neat: it gives P , the expected errors in parameter estimates, in terms of V , the expected errors in outputs, and Φ , the regression matrix. The inverse of P $P^{-1} = \Phi^T V^{-1} \Phi$ is the *Fisher information matrix* and describes information contents of the data. It “rises” when the variance in the observations falls or when more data become available.

It is seen from Eq. (1.29) that the parameter estimate covariance matrix depends on W , or more exactly on $W^T W$. As W contains the weights we assign to the prediction errors, we are “free” to choose it and the natural question is what choice of W , and hence $W^T W$, makes P smallest? The optimal choice turns out to be $W^T W = V^{-1}$ [15] and surprisingly

the resulting P is the same as when observations have the same “importance”, in Eq. (1.30). The parameter estimate corresponding to this choice is known as the *best linear unbiased estimate (BLUE)* or the *Markov estimate*.

1.4 SYSTEM REDUCTION BASED ON ERROR REDUCTION RATIO

The importance of eliminating redundant parameters from the model cannot be overstated in system identification. For a model class complex enough to contain the true description of the system, when the number of parameters increases in the model structure, so does the expected prediction error variance [16]. Therefore, there is a strict penalty associated with the presence of superfluous parameters in the model; the model must be as *parsimonious* as possible. Other than parsimony, there are at least two merits redeemed by system simplification: predefined fixed-point configuration and attractors that are topologically closer to the original ones [17].

We may connect observations, predictions, and prediction errors by noting that in least squares, prediction error and prediction are perpendicular; $\hat{Y}^T \mathbf{e} = 0$. As a result

$$\|Y\|_{\ell^2}^2 = (\hat{Y} + \mathbf{e})^T (\hat{Y} + \mathbf{e}) = \|\hat{Y}\|_{\ell^2}^2 + \|\mathbf{e}\|_{\ell^2}^2. \quad (1.31)$$

We define an error reduction ratio (*ERR*) equal to the ratio of 2-norm of predictions to observations,

$$ERR = \frac{\|\hat{Y}\|_{\ell^2}^2}{\|Y\|_{\ell^2}^2}. \quad (1.32)$$

It is obvious that $ERR = \frac{\|QR\hat{\theta}\|_{\ell^2}^2}{\|Y\|_{\ell^2}^2} = \frac{\|\beta\|_{\ell^2}^2}{\|Y\|_{\ell^2}^2} = \frac{\sum_j \beta_j^2}{\|Y\|_{\ell^2}^2} = \sum_j ERR_j$; ERR_j is the error

reduction caused by the presence of the j -th parameter in the model [18]. A threshold is

assumed for ERR_j below which the contribution of $\hat{\theta}_j$ to the error reduction is considered insignificant; we are allowed to omit that parameter. The parameters for the resulting “reduced-dimension” regression are then updated. This process is repeated until all remaining parameters give an ERR above the threshold. It should be pointed out that this system simplification procedure certainly yields a final parameter set with as many redundant parameters eliminated as possible but that set is not always “better” than a non-reduced system in all senses. Eliminating parameters is equivalent to solving the original least squares problem with some constraint equations. The original parameter estimates will be modified to take these constraints (eliminated parameters) into account. For finite differences, especially when data has some error, the constrained parameter estimates (i.e. system simplification) might have slightly more error than the unconstrained ones (i.e. without system simplification). This happens because the remaining terms are not accurate enough to yield the least prediction error with parameter values close to the true ones; some undesired changes in the estimates w.r.t the true parameter values are inevitable. Examples of this phenomenon are discussed in more detail with numerical experiments.

CHAPTER 2. SPECTRAL METHODS IN 1D, PERIODIC DOMAINS

2.1 HISTORY OF SPECTRAL METHODS

Since the days of Fourier, spectral expansions have been in use for *analytic* investigations of differential equations but the idea of spectral methods for *numerical* solutions can be traced back to the advent of algorithmic interpolation of functions by Lanczos as early as 1938 [19]. His work established that the accuracy of the solution by, what we now know as, spectral methods is crucially dependent on the appropriate selection of the trial functions and the distribution of the “grid points.” Before him, a variety of trial functions and *arbitrary* distributions of grid points were in use.

Years later in 1972, Orszag applied spectral methods to compute, with unprecedented accuracy, the Reynolds number corresponding to transition from laminar to turbulent for a certain flow. Kreiss and Oliger were also among the first to use spectral methods for different PDE solutions around the same time; their method – like most other spectral methods – took great advantage of the fast Fourier transform [20]. We should mention that Silberman had used spectral Galerkin methods in meteorological modeling in 1954. But it was not before Orszag (and others) had devised the transform method (circa 1970) that these methods became practical for nonlinear problems; still for very complicated nonlinear problems, spectral Galerkin methods on fine grids are not feasible [21].

2.2 FINITE DIFFERENCE VS. SPECTRAL METHODS: LOCAL VS. GLOBAL

One way of looking at spectral methods is to see them as infinite order finite differences. Generating finite difference formulas then has its own seemingly differing interpretations. One that sheds much light on the process is through the concept of interpolation: approximate the function by a *local* “polynomial” interpolant and get the derivative approximations by differentiating that interpolant. For a function that varies slowly, local interpolants are reasonable and in fact it makes little sense to use the function values far away from the point in question for estimating the derivative; the derivative is a *local* property of a function and it might seem unnecessary to use lots of function values away from the point of interest. It is a whole different ball game for functions with fast spatial/temporal variations. We *cannot* just use a finer grid to capture those variations accurately; that would require significant computational resources. We have to resort to alternatives that are successful with coarser grids and spectral methods are such methods; they are *global* meaning at any given point, they use *all* the information available from the entire domain. Furthermore, for “smooth” enough solutions, looking at the whole domain exploits that smoothness and offers “spectral” decay rates for approximation error. Contrariwise, local methods do not see past a certain number of points away from the point of interest and are therefore unable to see the smoothness, let alone exploit it.

2.3 SPECTRAL TRIAL FUNCTIONS AND TEST FUNCTIONS

Rigorously speaking, what we explained through interpolation is the outcome of a more general process involving trial functions (also known as expansion, basis or approximating functions) and test functions (also known as, weight functions) as spectral methods can be viewed as extreme cases of the class of discretization schemes for PDEs

known as the *method of weighted residuals* (MWR). Trial functions are used as bases for writing the solution as an expansion in a truncated series. Test functions are used to test the consistency of the scheme, i.e. it being satisfied, as “closely” as possible, by that expansion. As previously stated, spectral schemes are global; this means they are *based* on infinitely differentiable global functions which are tensor products of the eigenfunctions coming from *singular* Sturm-Liouville eigenproblems [22].

Spectral methods are distinguished by the type of the method (based on the choice of test functions) as well as by the particular choice of the trial functions. From the test functions point of view, we have three approaches,

- *Galerkin method* chooses them the same as the trial functions and they hence are infinitely smooth functions and each, separately, satisfies the boundary conditions. Solving the weak form (i.e., setting to zero the integral of the residual times each of the test functions) gives the approximate solution in this sense. In other words, Galerkin combines the original basis functions into a new set with all the functions satisfying the boundary conditions. Then it forces the residual to be orthogonal to as many of these new basis functions as possible.
- *tau methods* are similar to Galerkin methods regarding the weak form part. However, for tau methods test functions are independent of the boundary conditions and some supplementary equations are needed to take care of the boundaries.
- *Collocation methods* (also known as *pseudospectral methods*) have as their test functions the periodic Dirac delta functions shifted so as to be centered at the

“suitably” chosen grid points. This is equivalent to setting the residual to zero at those grid points.

The trial functions commonly in use are trigonometric polynomials, Chebyshev polynomials, and Legendre polynomials. The last two are the two more commonly used members of Jacobi polynomials whose members constitute a “zoo of polynomials”.

Thus from one viewpoint, the foundation of collocation spectral methods is then the spectral differentiation operators we get from differentiating these polynomials whose right selection depends on the domain at hand. On periodic domains, a uniform grid together with trigonometric polynomial interpolants (i.e. trigonometric polynomials as trial functions) leads to Fourier spectral methods.

2.4 PERIODIC DOMAINS AND FOURIER SPECTRAL METHODS

We deal with periodic boundaries in the first part of this work; they are the best choice for computation when the problem at hand is not related to the boundaries or when the coordinate space is periodic. Furthermore, the first spectral methods were used for simulations of homogeneous turbulence on *periodic domains*.

2.4.1 CONTINUOUS INFINITE AND FINITE FOURIER EXPANSIONS: TRUNCATION ERROR

We start with the fundamentals of Fourier spectral methods on a periodic domain, namely the continuous infinite Fourier expansion $F(u)$ of a function $u(x) \in L^2[-\pi, \pi]$

$$\begin{aligned} F[u] &= \sum_{|k| \leq \infty} \hat{u}_k e^{ikx}, \\ \hat{u}_k &= \frac{1}{2\pi} \int_{-\pi}^{\pi} u(x) e^{-ikx} dx. \end{aligned} \tag{2.1}$$

An infinite series is a theoretical concept rather than a practical tool and our goal then is to see how well the *truncated* continuous Fourier expansion

$$P_N u(x) = \sum_{|k| \leq N/2} \hat{u}_k e^{ikx} \quad (2.2)$$

approximates the function. Clearly, P_N is a projection operator from the infinite dimensional Hilbert space B to the finite dimensional space

$$\hat{B}_N = \text{span} \{ e^{ikx} \mid |k| \leq N/2 \} \quad \dim(\hat{B}_N) = N+1. \quad (2.3)$$

If the sum of squares of the coefficients is bounded we have convergence in L^2 norm and if, moreover, the sum of the absolute values of the coefficients is bounded we have convergence in L^∞ ; that is uniform convergence. It is concluded from the “convergence” of the *truncated* series that the error is (as it must) dominated by the *tail* of the series

$$\|u - P_N u\|_{L^2[-\pi, \pi]}^2 = 2\pi \sum_{|k| \geq N/2} |\hat{u}_k|^2, \quad (2.4)$$

and

$$\|u - P_N u\|_{L^\infty[-\pi, \pi]}^2 \leq \sum_{|k| \geq N/2} |\hat{u}_k|. \quad (2.5)$$

These both imply that the *truncation error depends solely on how fast the expansion coefficients decay*.

2.4.2 DIFFERENTIATION OF CONTINUOUS EXPANSION

Now that we know how close an approximation is the truncated expansion to the function, the natural question will be on the closeness of the derivatives, as we are to differentiate with the spectral operators in the end. For a sufficiently smooth function, term by term differentiation of the truncated expansion yields

$$\frac{d^p}{dx^p} P_N u(x) = \sum_{|k| \leq N/2} \frac{d^p}{dx^p} \hat{u}_k e^{ikx} = \sum_{|k| \leq N/2} (in)^p \hat{u}_k e^{ikx}. \quad (2.6)$$

The right-hand of Eq. (2.6) is $P_N \frac{d^p}{dx^p} u(x)$ which implies that the differentiation and continuous projection operators commute. More generally, for any constant coefficient differentiation operator L there is no truncation error

$$\begin{aligned} P_N L u &= P_N L P_N u, \\ \text{or} \\ P_N L (I - P_N) u &= 0. \end{aligned} \quad (2.7)$$

2.4.3 DISCRETE FOURIER EXPANSION: EVEN AND ODD QUADRATURES

The *continuous* Fourier expansion entails *analytical* evaluation of the integrals in Eq. (2.1) to get the expansion coefficients, a task that is neither plausible nor feasible in general. As a resort, we *approximate* those integrals using quadrature formulas, which depend on the exact position of grid points and whether we choose an even or odd number of grid points results in slightly different schemes.

Let $\{x_j\} \subset h\mathbb{Z}$ denote a set of grid points with h being their *uniform* spacing and choose $[0, 2\pi)$ as the fundamental periodic domain.

Even Expansion

We start with the case of an even number of grid points, N . The grid spacing is then $h = 2\pi / N$ and the x indices run as $j = 0, 1, \dots, N-1$. Furthermore suppose, for convenience, that $v = \{v_j\}$ is a spatial grid function in ℓ_N^2 space which equals the original function $u(x) \in C_p^0[0, 2\pi]$ on the grid; that is, $v_j = u(x_j)$. Applying the trapezoidal quadrature to Eq. (2.1) yields the discrete Fourier coefficients

$$\tilde{u}_k = \frac{1}{N} \sum_{j=0}^{N-1} u(x_j) e^{-ikx_j}, \text{ not ready to use, yet!} \quad (2.8)$$

The quadrature formula upon which Eq. (2.8) is predicated, namely,

$$\frac{1}{2\pi} \int_0^{2\pi} u(x) dx = \frac{1}{N} \sum_{j=0}^{N-1} u(x_j), \quad (2.9)$$

is exact for any trigonometric polynomial $u(x) = e^{ikx}$, $|k| < N$, i.e., for $u(x) \in \hat{B}_{2N-2}$.

We do a halving of the terms for the amplitude densities of the first and last mode – a step we will thoroughly explain shortly – to get the *discrete Fourier coefficients*

$$\begin{aligned} \tilde{u}_k &= \frac{1}{N\tilde{c}_k} \sum_{j=0}^{N-1} u(x_j) e^{-ikx_j} \\ \tilde{c}_k &= \begin{cases} 2 & |k| = N/2 \\ 1 & |k| < N/2 \end{cases}, \end{aligned} \quad (2.10)$$

whose substitution in Eq. (2.2) defines a new projection of u

$$I_N u(x) = \sum_{|k| \leq N/2} \tilde{u}_k e^{ikx} \quad (2.11)$$

Note from Eq. (2.10) that $\tilde{u}_{-N/2} = \tilde{u}_{N/2}$; we have exactly N independent Fourier coefficients, corresponding to the N quadrature points. Consequently, $I_N \sin(Nx/2) = 0$ and the function $\sin(Nx/2)$ cannot be represented by the discrete expansion Eq. (2.11).

So the finite dimensional space onto which I_N projects is $\tilde{B}_N = \hat{B}_N - \sin(Nx/2)$ or

$$\tilde{B}_N = \text{span} \{ (\cos(kx), 0 \leq k \leq N/2) \cup (\sin(kx), 1 \leq k \leq N/2 - 1) \} \quad (2.12)$$

whose dimension is N as opposed to $N+1$ for that of $\hat{B}_N$.

We have yet to discuss a point of paramount importance. Why did we choose the particular definition of the discrete expansion coefficients as given by Eq. (2.10)? Let us

look into that from one view point for now and dig deeper in the next section. Substitute the discrete coefficients of Eq. (2.10) in the discrete expansion of Eq. (2.11) to obtain,

$$I_N u(x) = \sum_{|k| \leq N/2} \left(\frac{1}{N \tilde{C}_k} \sum_{j=0}^{N-1} u(x_j) e^{-ikx_j} \right) e^{ikx} \quad (2.13)$$

or, after exchanging the order of summation,

$$I_N u(x) = \sum_{j=0}^{N-1} u(x_j) g_j(x), \quad (2.14)$$

$$g_j(x) = \sum_{|k| \leq N/2} \frac{1}{N \tilde{C}_k} e^{ik(x-x_j)} = \frac{\sin(N(x-x_j)/2)}{N \tan((x-x_j)/2)}.$$

It is clear that $g_j(x_i) = \delta_{ij}$ and that $g_j(x) \in \hat{B}_N$ – as $g_j(x)$ is a trigonometric polynomial of degree not more than $N/2$. It is also evident that there is no contribution from the term $\sin(Nx/2)$ in $g_j(x)$, whence $g_j(x) \in \tilde{B}_N$. This is an interesting consequence as it means $I_N u(x)$ interpolates $u(x)$ at the quadrature points of the trapezoidal formula. $g_j(x)$ is the *periodic sinc function*, the unique periodic *band-limited* interpolant to the discrete delta – band-limited in the sense that its Fourier transform has compact support contained in $[-N/2, N/2]$. Figure 2-1 depicts the periodic sinc function centered at 0 on an interval of $[0, 2\pi)$, with 9 sign changes; $g_j(x)$ is zero at all grid points except for $x = x_j$. Since it is periodic, this pattern repeats itself every $x = 2\pi$.

Figure 2-1 Periodic Sinc function $g_j(x)$ also known as the trigonometric interpolation polynomial

Figure 2-2 Non-periodic Sinc function

Figure 2-2 illustrates that the ordinary (i.e., non-periodic) sinc function damps with increasing x and that like the periodic sinc function, it is zero at the internal grid points.

This can be seen analytically by using the Taylor expansion of the sine near $x = 0$ which gives $\text{sinc}(x) \approx \frac{(Nx/2) - (Nx/2)^3/6 + (Nx/2)^5/120 - \dots}{Nx/2}$ which is zero at every grid

point other than the one at the origin as the root there is cancelled by the denominator.

To see the kinship between the trigonometric interpolation function and sinc function it suffices to Taylor expand the cotangent in Eq. (2.14) to obtain

$$\begin{aligned} g_j(x) &= \frac{\sin(N(x-x_j)/2)}{N(x-x_j)/2} \left\{ 1 - \frac{(x-x_j)^2}{12} - \frac{(x-x_j)^4}{720} + \dots \right\} \\ &= \text{sinc}(N(x-x_j)/2) \left\{ 1 - \frac{(x-x_j)^2}{12} - \frac{(x-x_j)^4}{720} + \dots \right\} \end{aligned} \quad (2.15)$$

This indicates that near its peak, the trigonometric interpolation function (periodic sinc function) and the ordinary sinc function cannot be told apart. Let us throw in this aside here that the trigonometric cardinal function is the result of replicating an infinite number of sinc functions supported at points one period (2π) apart and summing them up. This can be seen by mentally making a copy of Figure 2-2, flipping it to get the left part of the sinc not shown in that figure and then putting one on another and adding them. Needless to say, the result will be Figure 2-1.

Notice that without the halving of the first and last terms in $g_j(x)$, and sans the normalizing $1/N$, we get the *Dirichlet kernel* $D_N(x)$

$$D_N(x) = \sum_{|k| \leq N/2} e^{ikx} = \frac{\sin((N+1)x/2)}{\sin(x/2)}, \quad (2.16)$$

which can be seen as the (scaled) projection of the Dirac delta function onto the space spanned by the Fourier basis, as the continuous Fourier coefficients for that projection are

$$\hat{\delta}_k = \frac{1}{2\pi} \int_0^{2\pi} \delta(x) e^{-ikx} dx = \frac{1}{2\pi} \quad (2.17)$$

and hence the projection is

$$P_N \delta(x) = \sum_{|k| \leq N/2} \frac{1}{2\pi} e^{ikx} = \frac{1}{2\pi} D_N(x), \quad (2.18)$$

a scaled Dirichlet kernel. Figure 2-3 show that, unlike the periodic sinc function, the Dirichlet kernel is *not* zero at the internal grid points of every period.

Figure 2-3 Normalized Dirichlet kernel

Odd Expansion

On a grid with an odd number of points, $x_j = \frac{2\pi}{N+1}j, j \in \{0, 1, \dots, N\}$, the trapezoidal quadrature yields,

$$\tilde{u}_k = \frac{1}{N+1} \sum_{j=0}^N u(x_j) e^{-ikx_j}, \quad (2.19)$$

with the interpolation operator

$$I_N^{odd} u(x) = \sum_{|k| \leq N/2} \tilde{u}_k e^{ikx}. \quad (2.20)$$

Through the Lagrange interpolation polynomial the scheme is

$$\begin{aligned} I_N^{odd} u(x) &= \sum_{j=0}^N u(x_j) h_j(x), \\ h_j(x) &= \frac{\sin((N+1)(x-x_j)/2)}{(N+1)\sin((x-x_j)/2)}. \end{aligned} \quad (2.21)$$

It can be shown that $h_j(x) \in \hat{B}_N$, unlike $g_j(x) \in \tilde{B}_N$ [23]. Note that $h_j(x)$ is the normalized Dirichlet kernel - c.f. Eq. (2.16); no halving of the first and last amplitude densities is required for the case of an odd number of quadrature points.

2.5 INTERPOLATION APPROACH AND FOURIER PAIRS

There is another approach to reach the discrete Fourier expansion of a function, based on interpolation. Now that we have *proven* I_N (the discrete projection operator) to be an interpolation operator if we use the trapezoidal quadrature, the interpolation approach might seem obvious and unnecessary to be dealt with but we have, nonetheless, chosen to present it. For one thing, interpolation is usually introduced independently of the

projection. Furthermore, it sheds light on how we came up with the “magical” halving of the first and last amplitudes (discrete expansion coefficients).

We are going to take the discrete Fourier Transform pairs below as given. This might sound like putting the cart before the horse, and it is if looked at these pairs as coming from the discrete Fourier expansion of the previous section. However, that is not the only way to look at them!

There are a handful of pairs, differing in scale factors. To be consistent, our discrete Fourier Transform (DFT), which gives the discrete Fourier coefficients, would be

$$\tilde{u}_k = \hat{v}_k = \frac{1}{N} \sum_{j=0}^{N-1} v_j e^{ikx_j}, \quad (2.22)$$

and its inverse, returning the grid function, would be

$$I_N u(x_j) = v_j = \sum_{k=-N/2}^{N/2-1} \hat{v}_k e^{ikx_j}, \quad j = 0, 1, \dots, N-1. \quad (2.23)$$

The gist of the interpolation approach is to consider the periodic Kronecker delta δ_j ,

$$\delta_j = \begin{cases} 1 & j \equiv 0 \pmod{N} \\ 0 & j \not\equiv 0 \pmod{N} \end{cases}. \quad (2.24)$$

and notice that we can expand any grid function v in the basis of the shifted periodic Kronecker delta,

$$v_j = \sum_{m=-N/2}^{N/2-1} v_m \delta_{j-m}. \quad (2.25)$$

Now if we find a “suitable” interpolant to the *Kronecker delta*, we will then have a “suitable” interpolant to *any* periodic grid function.

2.6 PERIODIC SINC FUNCTION

To get an interpolant, we evaluate the inverse DFT, Eq.(2.23), not just on the grid points x_j but on *every* point x in the domain. That does the trick,

$$I_N u(x) = \sum_{k=-N/2}^{N/2-1} \hat{v}_k e^{ikx}, x \in [0, 2\pi) \text{ not correct, yet!} \quad (2.26)$$

There is a problem, however, if we use Eq. (2.26) as it stands. It treats the highest wavenumbers “discriminately”. This can be seen by evaluating Eq. (2.26) which would produce the “lonesome” term $\hat{v}_{-N/2} e^{i(-N/2)x}$, coming from the highest wavenumber, with the derivative $\hat{v}_{-N/2} i(-N/2) e^{i(-N/2)x}$. The lonesome term is a real sawtooth wave on the grid, $e^{i(-N/2)x_j} = e^{i(-\pi/h)jh} = e^{-ij\pi}$, oscillating between 1 or -1 for even and odd j ’s, respectively. Therefore, on the grid, the aforementioned derivative must be real (and actually zero) which is not the case, as seen from the coefficient $\hat{v}_{-N/2} i(-N/2)$. We can fix this by noting that the exponentials are not changed if any integer multiple of N is added to k . Thus, we may interpret $\hat{v}_{N-k}$ as $\hat{v}_{-k}$, $k = 1, 2, \dots, N/2 - 1$ and write $\hat{v}_{N/2}$ as $\hat{v}_{\pm N/2}$, seeing it as coming from equal amounts of the modes $N/2$ and $-N/2$. In other words, we are bringing $\hat{v}_{N/2} e^{i(N/2)x}$ into the summation to kill the loneliness of $\hat{v}_{-N/2} e^{i(-N/2)x}$. Doing so yields $\hat{v}_{-N/2} (e^{i(N/2)x} + e^{i(-N/2)x})$. We must, however, divide this by two since we just had one of these two equal terms originally

$$I_N u(x) = \sum_{k=-N/2+1}^{N/2-1} \hat{v}_k e^{ikx} + \hat{v}_{N/2} \underbrace{\left(\frac{e^{i(N/2)x} + e^{-i(N/2)x}}{2} \right)}_{\cos(Nx/2)}, x \in [0, 2\pi). \quad (2.27)$$

If we embed the halving of the first and last terms in the summation notation by indicating it with a double prime, we are eventually left with what we set out to achieve, the trigonometric interpolation formula,

$$I_N u(x) = \sum_{|k| \leq N/2}'' \hat{v}_k e^{ikx}, \quad x \in [0, 2\pi). \quad (2.28)$$

Applying this to the Kronecker delta, we arrive at Eq. (2.14), this time via interpolation.

2.7 PERIODIC DIFFERENTIATION TOOLS: MATRICES AND EXPANSION COEFFICIENTS

Spectral differentiation is a linear operation on ℓ_N^2 ; therefore its operator must be representable by a matrix. An approximation to the derivative of $u(x)$ at the quadrature points is obtained by differentiating the interpolation,

$$\frac{d}{dx} I_N u(x) \big|_{x_i} = \sum_{j=0}^{N-1} u(x_j) \frac{d}{dx} g_j(x) \big|_{x_i} = \sum_{j=0}^{N-1} D_{ij} u(x_j), \quad (2.29)$$

where

$$\frac{d}{dx} g_j(x) = \frac{\cos(Nx/2)}{2 \tan(x/2)} - \frac{\sin(Nx/2)}{2N \sin^2(x/2)}, \quad (2.30)$$

and

$$D_{i,j} = \frac{d}{dx} g_j(x) \big|_{x_i} = \begin{cases} 0 & i = j, \\ \frac{(-1)^{i+j}}{2} \cot\left(\frac{x_i - x_j}{2}\right) & i \neq j. \end{cases} \quad (2.31)$$

From Eq. (2.31) it can be verified that the operator D is a *Toeplitz* matrix depending only on $i-j$ entries. In fact, it is more; it is *circulant* since the entries depend on $(i-j) \bmod(N)$ only. The matrix, written in open form, is,

$$D = \begin{bmatrix} 0 & & & & & -\cot(1h/2) \\ -\cot(1h/2) & & & & & \cot(2h/2) \\ & \ddots & & & & -\cot(3h/2) \\ -\cot(3h/2) & & & & & \vdots \\ -\cot(2h/2) & & \ddots & \ddots & & \ddots & \cot(1h/2) \\ \cot(1h/2) & -\cot(2h/2) & \cdots & -\cot(1h/2) & & 0 \end{bmatrix}, \quad (2.32)$$

and the *first* derivative approximation is given by multiplying this matrix to the vector of samples. This matrix is depicted in Figure 2-4.

Figure 2-4 16-by-16 Fourier differentiation matrix. Skew symmetry is readily noticed in this figure.

Figure 2-5 Eigenvalues of 16-by-16 Fourier differentiation matrix. *Top:* Axes with non-equal scaling to show the effect of double-precision roundoff. *Bottom:* Axes with equal scaling show how the true eigenvalues are distributed; they all lie on the imaginary axis. Note that in case of an even quadrature grid, like here, the zero eigenvalue has an algebraic multiplicity of 2.

For the second derivative operator, a second differentiation of the interpolation gives,

$$D_{i,j}^{(2)} = \frac{d^2}{dx^2} g_j(x) \big|_{x_i} = \begin{cases} -\frac{N^2 + 2}{12} & i = j, \\ -\frac{(-1)^{i+j}}{2} \csc^2\left(\frac{x_i - x_j}{2}\right) & i \neq j. \end{cases} \quad (2.33)$$

Written explicitly, this matrix is

$$D^{(2)} = \begin{bmatrix} -\frac{N^2+2}{12} & & & \frac{1}{2} \csc^2\left(\frac{1h}{2}\right) \\ \frac{1}{2} \csc^2\left(\frac{1h}{2}\right) & & & -\frac{1}{2} \csc^2\left(\frac{2h}{2}\right) \\ & \vdots & & \frac{1}{2} \csc^2\left(\frac{3h}{2}\right) \\ \frac{1}{2} \csc^2\left(\frac{3h}{2}\right) & & & \vdots \\ -\frac{1}{2} \csc^2\left(\frac{2h}{2}\right) & \ddots & \ddots & \frac{1}{2} \csc^2\left(\frac{1h}{2}\right) \\ \frac{1}{2} \csc^2\left(\frac{1h}{2}\right) & -\frac{1}{2} \csc^2\left(\frac{2h}{2}\right) & \cdots & \frac{1}{2} \csc^2\left(\frac{1h}{2}\right) & -\frac{N^2+2}{12} \end{bmatrix} \quad (2.34)$$

In the case of even number of quadrature points, $D^{(2)}$ is not equal to D^2 ; to wit,

$$I_N \frac{d^2}{dx^2} I_N \neq \left(I_N \frac{d}{dx} \right)^2 I_N \quad (2.35)$$

Let us show that for $\cos(Nx/2)$:

$$I_N \frac{d^2}{dx^2} I_N \cos(Nx/2) = I_N \left(-\left(\frac{N}{2}\right)^2 \cos(Nx/2) \right) = -\left(\frac{N}{2}\right)^2 \cos(Nx/2).$$

Contrariwise, $\left(I_N \frac{d}{dx} \right)^2 I_N \cos(Nx/2) = \left[I_N \left(-\frac{N}{2} \sin(Nx/2) \right) \right]^2 = 0$ and the two are not

the same. The differentiation operator based on an even number of points takes elements

of $\tilde{B}_N$ out of $\tilde{B}_N$. Elements of $\hat{B}_N$, however, stay in $\hat{B}_N$ after differentiation and thus for differentiation based on an odd number of quadrature points we have

$$I_N^{odd} \frac{d^2}{dx^2} I_N^{odd} = \left(I_N^{odd} \frac{d}{dx} \right)^2 I_N^{odd}, \quad (2.36)$$

or more generally $D^{odd(q)} = (D^{odd})^q$, where D^{odd} is the differentiation operator based on an odd number of quadrature points.

As mentioned above, D is a circulant matrix and a circulant matrix signals a convolution on a periodic physical domain, or equivalently a pointwise multiplication in the Fourier domain. This insight together with Eq. (2.11) suggests another approach to do the spectral differentiation. Instead of dwelling in the physical domain (which imposes a convolution) we go to the Fourier domain and seek that equivalent pointwise multiplication. To do that, we note that Eq. (2.27) is dealing with the data in the Fourier domain. Let us take the first derivative to get,

$$\frac{d}{dx} I_N u(x) = \sum_{k=-N/2+1}^{N/2-1} ik \hat{v}_k e^{ikx} - \frac{N}{2} \hat{v}_{N/2} \sin\left(\frac{N}{2}x\right), x \in [0, 2\pi), \quad (2.37)$$

and the second derivative to get,

$$\frac{d^2}{dx^2} I_N u(x) = \sum_{k=-N/2+1}^{N/2-1} (ik)^2 \hat{v}_k e^{ikx} - \left(\frac{N}{2}\right)^2 \hat{v}_{N/2} \cos\left(\frac{N}{2}x\right), x \in [0, 2\pi) \quad (2.38)$$

Notice that we have partially opened the summation, and taken out the two terms corresponding to the first and last modes, so that it is seen that, at the quadrature points, $x = x_j = jh = j2\pi/N$, the modes corresponding to $k = 0$ and $\pm N/2$ do not play a part in the first derivative (Eq. (2.37)), or any other derivative of odd order for that matter. In the second derivative (Eq. (2.38)), or other derivatives of even order, however, only the $k = 0$ mode is “barren”.

Therefore, the second approach is to first transform the grid function v from the physical space to the Fourier space by a complex DFT and get $\hat{v}$. In the latter space, as seen from Eq.’s (2.37) and (2.38), differentiation is essentially a multiplication of every $\hat{v}_k$ by its corresponding ik , with special attention to $k = \pm N/2$ for derivatives of odd order. To

have the derivative in the physical space, the result is then transformed back by an inverse DFT. DFT and its inverse can be handled by FFT algorithms efficiently.

2.7.1 EXPANSION COEFFICIENTS AND ORDER OF WAVENUMBERS: MATLAB IMPLEMENTATION

There is a bit of technicality regarding a discrepancy in the order of wavenumbers used by MATLAB and us. MATLAB, which we are using as our numerical experiment “lab”, returns $\hat{v}_k$ ’s with the following order of wavenumbers in its implementation of the FFT,

$$0, 1, \dots, \frac{N}{2}-1, \frac{N}{2}, \frac{N}{2}+1, \dots, N-1 \quad (2.39)$$

which, due to periodicity, is equivalent to

$$0, 1, \dots, \frac{N}{2}-1, \frac{N}{2}-N, \frac{N}{2}+1-N, \dots, N-1-N, \quad (2.40)$$

or

$$0, 1, \dots, \frac{N}{2}-1, -\frac{N}{2}, -\frac{N}{2}+1, \dots, -1. \quad (2.41)$$

Thus, we must multiply the output elements of MATLAB’s FFT by k ’s according to Eq. (2.41), noting that, with an even number of quadrature points, the reddened element should be treated differently for odd and even derivatives as mentioned before.

In a nutshell, n th order spectral differentiation, before reverting to the physical domain, amounts to multiplying the grid function by the appropriate vector below,

$$(ik)_{\text{ew}}^n = i^n \left[0, \left(1, \dots, \left(\frac{N}{2}-1 \right) \right), \mathbf{0}, -\left(\left(\frac{N}{2}-1 \right), \dots, 1 \right) \right] \cdot \wedge n, \text{ (odd order deriv. } n), \quad (2.42)$$

$$(ik)_{\text{ew}}^n = i^n \left[0, \left(1, \dots, \left(\frac{N}{2} - 1 \right) \right), \frac{N}{2}, - \left(\left(\frac{N}{2} - 1 \right), \dots, 1 \right) \right] .^{\wedge} n, \text{ (even order deriv. } n), \quad (2.43)$$

where the subscript *ew* and the operator $.^{\wedge}$ (dot hat) emphasize that raising to power is implemented element-wise, the only way it makes sense for a vector.

Concisely,

$$\mathbf{v}^{(n)} = \underbrace{\mathbf{F}_N^{-1} (ik)_{\text{ew}}^n}_{\mathbf{w}^{(n)}} \times \underbrace{\mathbf{F}_N \mathbf{v}}_{\hat{\mathbf{v}}}, \quad (2.44)$$

where $\mathbf{v}^{(n)}$ is the spectral approximation to the n th order derivative of $u(x)$ at the quadrature points x_j , $\mathbf{F}_N$ and $\mathbf{F}_N^{-1}$ denote the DFT and inverse DFT matrices implemented by `fft()` and `ifft()` respectively, and $. \times$ asks for an element-wise multiplication.

2.8 CONTINUOUS MINUS DISCRETE FOURIER EXPANSION: ALIASING AND HOW TO AVOID IT DURING SAMPLING

On a grid of quadrature points x_j , the k -th mode looks like the $(k + mN)$ -th mode; they are indistinguishable: $e^{i(k+mN)x_j} = e^{ikx_j} e^{i2\pi m} = e^{ikx_j}$. As a result, each discrete Fourier coefficient contains the contribution of not only its continuous counterpart, but also of all other continuous coefficients that look equivalent on the grid; to wit,

$$\tilde{c}_k \tilde{u}_k = \hat{u}_k + \sum_{\substack{|m| \leq \infty \\ m \neq 0}} \hat{u}_{k+mN}, \quad (2.45)$$

which is derived by substituting the continuous Fourier expansion into the discrete expansion and using the Cauchy-Schwarz inequality. This phenomenon is known as aliasing. The finite continuous Fourier expansion and discrete Fourier expansion both

contain only modes $|k| \leq N/2$. In consequence, their difference, the aliasing error, $A_N u(x) = P_N u(x) - I_N u(x)$, can contain those modes only. We will later use this to derive the approximation error for discrete expansion.

In our problem, we have taken $k = -N/2, -N/2+1, \dots, N/2$ as the fundamental set of wavenumbers. Clearly the highest mode attainable by this set of wavenumbers is

$$\begin{aligned} k_{\text{critical}} &= (N/2) = \pi/h, \text{ or} \\ f_{\text{critical}} &= k_{\text{critical}} / (2\pi) = 1/(2h) = (1/2)f_{\text{sampling}}, \end{aligned} \quad (2.46)$$

where $f_{\text{sampling}} = 1/h$ and f_{critical} is known as the Nyquist critical frequency. Eq. (2.46) implies that to avoid aliasing, the sampling the frequency must be (at least) twice faster than the highest frequency present in the data. In other words, we must pick (at least) two samples in the smallest period present in the data, or aliasing will occur. As we have a finite discrete Fourier expansion, we can keep only a finite number of modes. Now for a function that is not bandwidth limited to $|k| < k_{\text{critical}}$ (that is, a function whose infinite continuous Fourier series is different from its finite (truncated) continuous Fourier series approximation), any continuous Fourier amplitude outside that band is, therefore, spuriously “folded over” into that band and so its effect appears in the equivalent discrete Fourier amplitude, according to Eq. (2.45) [24]. For inverse problems, aliasing becomes a concern particularly when we deal with noise-contaminated data. Noise creates high frequency components in the data and undersampling results in those components to be aliased to the low frequency ones. It is, therefore, standard practice to filter out the unwanted high frequencies before doing any sampling of a continuous signal. Moreover, variable-coefficient terms or nonlinear terms can give rise to high frequency modes and hence lead to aliasing [20]. Figure 2-6 and Figure 2-7 depict the effect of aliasing on the

appearance of the sine function. Figure 2-6 is a dot plot of $\sin(1/3000)$ whereas Figure 2-7 is $\sin(1/1000)$. None looks like a “sine” wave because, in both, “higher” frequencies are aliased by the eyes to “lower” frequencies.

Figure 2-6 Aliasing at work: $\sin(1 : 3000)$.

Figure 2-7 Aliasing at work: $\sin(1:1000)$.

2.9 APPROXIMATION ERROR OF FOURIER SPECTRAL METHODS: TRUNCATION PLUS ALIASING

Consistency is one of the fundamental elements of the convergence theory and it stipulates that the truncation error vanish as the number of terms used in the approximation approaches infinity. As the truncation error, $P_N L(I - P_N)u$, depends on the Fourier approximations not only of the function but its derivatives too, and since we are to compute derivatives using the Fourier spectral method, we use the Sobolev q -norm $H_p^q[0, 2\pi]$ to measure the smoothness of the derivatives in addition to the function,

$$\|u\|_{H_p^q[0, 2\pi]}^2 = \sum_{m=0}^q \int_0^{2\pi} |u^{(m)}(x)|^2 dx. \quad (2.47)$$

For periodic functions (to which the subscript p is alluding) an equivalent norm is defined in the mode space that allows for non-integer values of q too [23],

$$\|u\|_{W_p^q[0,2\pi]}^2 = \sum_{|k| \leq \infty} (1+k^{2q}) |\hat{u}|^2. \quad (2.48)$$

It can be shown that for any real r and any real q where $0 \leq q \leq r$, if $u(x) \in W_p^r[0,2\pi]$, then there exists a positive constant C (independent of N) such that

$$\|u - P_{2N}u\|_{W_p^q[0,2\pi]} \leq C \|u\|_{W_p^r[0,2\pi]} N^{-(r-q)}, \quad (2.49)$$

with the important implication that the smoother the function and its derivatives, the larger the value of r , and the better the approximation. This rate of convergence is referred to as *spectral convergence* and it is in contrast to local methods (e.g. finite differences or finite elements) where the rate of convergence is fixed and is independent of the smoothness of the function. In L^2 the bound simplifies to

$$\|u - P_{2N}u\|_{L^2[0,2\pi]} \leq C \|u^{(q)}\|_{L^2[0,2\pi]} N^{-q}, \quad (2.50)$$

for any function $u(x) \in H_p^q[0,2\pi]$, and if $u(x)$ is further analytic, using Stirling's formula $q! \sim q^q e^{-q}$ in $\|u^{(q)}\|_{L^2[0,2\pi]} \leq Cq! \|u\|_{L^2[0,2\pi]}$ and assuming that $q \propto N$ yields [23]

$$\|u - P_{2N}u\|_{L^2[0,2\pi]} \leq C \|u\|_{L^2[0,2\pi]} e^{-cN}, \quad (2.51)$$

which cues *exponential convergence for analytic functions*.

We are only one step away from stating the approximation error for the discrete Fourier expansion. We have to add the aliasing error to the truncation error as

$$\begin{aligned} \|u - I_{2N}u\|_{W_p^q[0,2\pi]}^2 &= \|(u - P_{2N}u) + (P_{2N}u - I_{2N}u)\|_{W_p^q[0,2\pi]}^2 \\ &\leq \|u - P_{2N}u\|_{W_p^q[0,2\pi]}^2 + \|P_{2N}u - I_{2N}u\|_{W_p^q[0,2\pi]}^2 \end{aligned} \quad (2.52)$$

We have already given a bound for the first component of Eq. (2.52), namely the truncation error. The second component, namely the *aliasing error*, is of the same order

as the truncation error, as long as the function has at least half a derivative. In consequence, the approximation error for the Fourier spectral method is

$$\|u - I_{2N}u\|_{W_p^q[0,2\pi]}^2 \leq C \|u\|_{W_p^r[0,2\pi]} N^{-(r-q)}. \quad (2.53)$$

We may justify these results, heuristically, by noting that a “smooth” function in the physical space decays rapidly in the Fourier space because “smoothness” translates to slow changes in the former space and since large wavenumbers correspond to fast changes, such a function has “little” of them. As a result, for such a function, the discretization errors are “small” since they are caused by the aliasing of higher wavenumbers to lower ones [25]. This is exactly the statement of the sampling theorem [24]. A numerical experiment, Figure 2-8, shows the considerable difference in accuracy, between finite differences and Fourier spectral methods, when dealing with the 1st to 4th derivatives of $u(x) = \sin(x) \exp(\cos(x))$. The finite differences employed in the experiment are the centered 4th order differences.

Figure 2-8 Comparison of infinity norms of 1st to 4th order finite differences and spectral differentiation errors with different number of samples for the function $\sin(x)\exp(\cos(x))$. **(a)** Fourier spectral differentiation and “spectral” rate of convergence **(b)** Finite differences and fixed (polynomial) rate of convergence of $O(h^4)$ and slower. Note that in (a) as the error reaches machine epsilon, (typically around 10^{-16} on double precision computers but maybe somewhat larger, perhaps by a factor of 100) roundoff takes over and the error fluctuates randomly about that value; this part of the graph is hence dubbed “roundoff plateau”.

CHAPTER 3. SPECTRAL METHODS IN 1D, NON-PERIODIC DOMAINS

3.1 NON-PERIODIC DOMAINS AND POLYNOMIAL SPECTRAL METHODS

The focus was on periodic problems and trigonometric (Fourier) spectral methods in the previous chapter. But not all problems are periodic. For non-periodic problems the first idea that might come to mind is periodic extension: to pretend that the problem is periodic and still use the trigonometric functions as the global basis functions. Unless the solution is exponentially close to a constant near the boundaries (or has several continuous derivatives there), this periodic treatment suffers from the global contamination caused by the discontinuities, Gibbs phenomenon in short. Without high order periodicity, the spectral convergence rate of Fourier methods plummets to that of local methods. The next idea is to replace the trigonometric functions with *algebraic (non-periodic) polynomials*; after all, if the function or any of its derivatives are not periodic, it does not make sense to use a periodic-basis expansion to represent it. For collocation methods, this change from trigonometric to algebraic cannot solve the problem by itself unless a quadrature is done on a specially *clustered grid* instead of a uniform one. For their attractive convergence features as well as abundance of literature that has studied them in detail, we use eigensolutions to *singular* Sturm-Liouville problems as our global basis.

3.2 ORTHOGONAL POLYNOMIALS AND SINGULAR STURM-LIOUVILLE EIGENPROBLEMS

Our focus will be on the finite interval $[-1,1]$. In the framework of equilibrium equations [15], a Sturm-Liouville problem is an extension to the ordinary $A^T C A \phi = \lambda w \phi$ which,

with $A = d/dx$ and $C = p(x)$ yields $-\frac{d}{dx} \left(p(x) \frac{d}{dx} \right) \phi(x) = \lambda w \phi(x)$. The extension

required is due to the presence of a linear term $q\phi$ in the Sturm-Liouville equation, which

stipulates $A = \begin{bmatrix} d/dx \\ I \end{bmatrix}$ and $C = \begin{bmatrix} p(x) & 0 \\ 0 & q(x) \end{bmatrix}$ to give the Sturm-Liouville operator

$$L\phi(x) = \begin{bmatrix} -d/dx & I \end{bmatrix} \begin{bmatrix} p(x) & 0 \\ 0 & q(x) \end{bmatrix} \begin{bmatrix} d/dx \\ I \end{bmatrix} \phi(x) = \lambda w(x) \phi(x), \quad (3.1)$$

accompanied by the “right” boundary conditions

$$\begin{aligned} \alpha_- \phi(-1) + \beta_- \phi'(-1) &= 0 & \alpha_-^2 + \beta_-^2 &\neq 0, \\ \alpha_+ \phi(1) + \beta_+ \phi'(1) &= 0 & \alpha_+^2 + \beta_+^2 &\neq 0. \end{aligned} \quad (3.2)$$

These boundary conditions are crucial as they make this operator self-adjoint. The digression we had to equilibrium frameworks makes this point heuristically clear; L was shown to be of the form $A^T C A$ which is clearly self-adjoint. Consequently, the eigenvalues λ_n are all real and the eigenfunctions ϕ_n form an orthogonal $L_w^2[-1,1]$ -complete basis. Rigorously, mutual orthogonality is

$$(\phi_m, \phi_n)_{L_w^2} = \int_{-1}^1 \phi_m(x) \phi_n(x) w(x) dx = \gamma_n \delta_{mn}, \quad \gamma_n = (\phi_n, \phi_n)_{L_w^2}. \quad (3.3)$$

completeness is when for any function $u(x) \in L_w^2[-1,1]$ we have

$$u(x) = \sum_{n=0}^{\infty} \hat{u}_n \phi_n(x), \quad (3.4)$$

with the natural continuous approximation

$$P_N u(x) = \sum_{n=0}^N \hat{u}_n \phi_n(x), \quad (3.5)$$

$$\hat{u}_n = \frac{1}{\gamma_n} (u, \phi_n)_{L_w^2[-1,1]}$$

Just as in the trigonometric expansion, P_N is a projection operator here too with the projection space being

$$B_N = \text{span} \{x^n\}_{n=0}^N \quad \dim(B_N) = N+1. \quad (3.6)$$

Similar to the trigonometric expansion, Parseval's identity implies that the decay of the truncation error depends on the decay of the expansion coefficients which, after simple algebraic manipulations, may be written as [26]

$$\hat{u}_n = \frac{1}{\gamma_n \lambda_n} \left[p(u' \phi_n - u \phi_n') \right]_{-1}^1 + \frac{1}{\gamma_n \lambda_n} \int_{-1}^1 [Lu(x)] \phi_n(x) dx \quad (3.7)$$

Here is where the *singularity* chips in for expansion based on algebraic polynomials; $p(\pm 1) = 0$ and that kills off the boundary term in Eq. (3.7) and repeated integration by

parts of the other term yields $\hat{u}_n \simeq C \frac{1}{(\lambda_n)^m} \left\| \left(\frac{L}{w(x)} \right)^m u(x) \right\|_{L_w^2[-1,1]}$, which implies spectral

decay, independent of the boundary conditions on u .

As our derivation of Eq. (3.7) did not rely upon the use of any specific basis, we must be able to see how we get spectral decay for trigonometric expansions too. It is true that Fourier basis comes from the solution of a *regular* Sturm-Liouville problem, and hence the boundary term cannot be wiped off by p being zero at the boundary points, but now *periodicity* chips in and causes the boundary term to disappear. The point is one way or

another the boundary term must go or else it creates a non-negligible contribution to the truncation error.

3.2.1 EVALUATING CHEBYSHEV POLYNOMIALS USING THREE-TERM RECURRENCE RELATIONS

With $p(x) = (1-x)^{\alpha+1}(1+x)^{\beta+1}$, $w(x) = (1-x)^\alpha(1+x)^\beta$, and $q(x) = cw(x)$, $\alpha, \beta > -1$, the eigensolutions to the singular Sturm-Liouville problem are polynomials of degree n and are specifically known as Jacobi polynomials. Obviously, with different pairs of α, β we will have different Jacobi polynomials and the question is which of them to choose to get the “best” finite expansion on the finite interval at hand. As alluded to by the quotation marks, the answer depends on the meaning of best we have in mind for a particular task, that is, on what norm we measure the error in. A subclass of Jacobi polynomials known as *ultraspherical polynomials* have $\alpha = \beta$ and a different normalization:

$$P_n^\alpha(x) = \frac{\Gamma(\alpha+1)\Gamma(n+2\alpha+1)}{\Gamma(2\alpha+1)\Gamma(n+\alpha+1)} P_n^{(\alpha,\alpha)}(x). \text{ With } \alpha = \beta = 0, w(x) = 1, \text{ and } q(x) = 0 \text{ we get}$$

Legendre polynomials $P_n(x)$ that are clearly orthogonal in $L^2[-1,1]$; they offer finite expansions that are best when the truncation error is measured in $L^2[-1,1]$ norm. We, however, opt for *Chebyshev polynomials* $T_n(x)$ given with $p(x) = \sqrt{1-x^2}$, $w(x) = 1/\sqrt{1-x^2}$, and $q(x) = 0$. The rationale behind this choice is that for Chebyshev polynomials we have a multitude of *close-form* expressions and relations. Furthermore the conversions between Chebyshev coefficients and the function values at the quadrature nodes can be quickly done by the fast cosine transform. Lastly, quadrature based on Chebyshev polynomials makes the discrete expansion (interpolation) more robust [20].

As an aside, attention is drawn to the point that Chebyshev polynomials constitute the “best” basis when the error is measured in the maximum norm. Figure 3-1 shows a number of Chebyshev polynomials on the fundamental interval. Note in the figure that, like all other Jacobi polynomials, Chebyshev polynomials have all their zeros in the fundamental interval.

Figure 3-1 Plot of Chebyshev polynomials of degree $n = 0, 2, 4, 7, 10, 15, 20$ on the fundamental interval $[-1, 1]$.

Chebyshev polynomials could be better understood if we expose the equivalence between Chebyshev series in x on $[-1, 1]$, Fourier series in θ on $\mathbb{R}$, and Laurent series in z on the unit circle. The relation between the independent variables of these series is

$$x = \operatorname{Re}(z) = \frac{z + z^{-1}}{2} = \cos(\theta). \text{ The transformation } x = \cos(\theta) \text{ changes the Sturm-Liouville}$$

problem of Chebyshev polynomials to the constant coefficient problem

$$\frac{d^2 T_n(\theta)}{d\theta^2} + \lambda_n T_n(\theta) = 0, \quad (3.8)$$

with the eigenvalues $\lambda_n = n^2$ and eigenfunctions $T_n(\theta) = \cos(n\theta)$. As a result, Chebyshev polynomials are

$$T_n(x) = \cos(n \arccos x) = \operatorname{Re} z^n = \frac{z^n + z^{-n}}{2} = \cos(n\theta). \quad (3.9)$$

The complex representation makes short work of finding the recurrence relations. As an example, here is the derivation of the *three-term recurrence relation for Chebyshev polynomials*

$$\begin{aligned} T_{n+1}(x) &= \frac{z^{n+1} + z^{-n-1}}{2} = \frac{z^n + z^{-n}}{2} (z + z^{-1}) - \frac{z^{n-1} + z^{1-n}}{2}, \\ &= 2xT_n(x) - T_{n-1}(x). \end{aligned} \quad (3.10)$$

3.3 DIFFERENTIATION OF CONTINUOUS CHEBYSHEV EXPANSION: TRIDIAGONAL RECURRENCE RELATIONS

The most important recurrence relation for our purposes would relate the expansion coefficients and those of its derivatives. Going from the expansion coefficients to those of its derivative is straightforward; we need to use the *three-term recursion for the first derivative*,

$$\frac{T'_{n+1}(x)}{n+1} = 2T_n(x) + \frac{T'_{n-1}(x)}{n-1}, \quad (3.11)$$

in the q -th derivative expansion, $\frac{d^{(q)}}{dx^{(q)}} u(x) = \sum_{n=0}^{\infty} \hat{u}_n \frac{d^{(q)}}{dx^{(q)}} T_n(x)$, to get the *tridiagonal*

integration operator

$$\hat{u}_n^{(q-1)} = \frac{c_{n-1}}{2n} \hat{u}_n^{(q)} - \frac{1}{2n} \hat{u}_{n+1}^{(q)} \quad (3.12)$$

Our problem, however, is actually the opposite. We need a *derivative operator* which we obtain by inverting the integral operator of Eq. (3.12)

$$c_{n-1}\hat{u}_{n-1}^{(q)} = \hat{u}_{n+1}^{(q)} + 2n\hat{u}_n^{(q-1)}. \quad (3.13)$$

3.4 GAUSS QUADRATURES AND DISCRETE INNER PRODUCTS

Although we now have recurrence relations giving the expansion coefficients of the derivative from those of the function, the very task of calculating the latter might be formidable as it entails evaluation of integrals analytically. For the case of trigonometric expansion the integrals giving coefficients were approximated using trapezoidal quadrature formulas; they were based on uniformly spaced quadrature points. For the current case of polynomial expansion, however, a quadrature of the form

$$\int_{-1}^1 r_N(x)w(x)dx = \sum_{j=0}^N r_N(x_j)w_j \quad (3.14)$$

is fitting. Independently of how the nodes x_j are distributed, this equation can be required to hold for $u(x)=1, x, \dots, x^N$, resulting in a linear system with a Van der Monde coefficient matrix which is nonsingular. Thus the weights w_j can always be found in such a way as to make Eq. (3.14) hold for all polynomials of degree up to N . The natural question now is if we can do better. It is reasonable to expect to be able to do better because up to here we have not used the degrees of freedom made available to us by the nodes. Gauss quadrature formulas exploit this fact and make the approximation exact for polynomials of much higher orders. There are three basic types of Gauss quadrature formulas, namely U Gauss-Lobatto, U Gauss-Radau, and U Gauss, where U is the ultraspherical polynomial based on which the quadrature points x_j are calculated. U -

Gauss-Lobatto includes both end points $x = \pm 1$ and the internal points are the roots of $\frac{d}{dx} P_N^{(\alpha)}(x)$. It is exact for $p(x) \in B_{2N-1}$. U -Gauss-Radau includes only one of the end points and the remaining points are the roots of $P_{N+1}^{(\alpha)}(x) + a_N P_N^{(\alpha)}(x)$. It is exact for $p(x) \in B_{2N}$. U Gauss includes none of the end points and its quadrature points are given by the roots of $P_{N+1}^{(\alpha)}(x)$. It is exact for $p(x) \in B_{2N+1}$. As an example let us see where this bonus exactness ($2N+1$ instead of N) is coming from for the U -Gauss quadrature. Any polynomial $u_{2N+1}(x)$ can be written as $u_{2N+1}(x) = s_N(x) P_{N+1}^{(\alpha)}(x) + r_N(x)$. Multiplying by $w(x)$ and integration over $[-1, 1]$ gives

$$\int_{-1}^1 u_{2N+1}(x) w(x) dx = \int_{-1}^1 s_N(x) P_{N+1}^{(\alpha)}(x) w(x) dx + \int_{-1}^1 r_N(x) w(x) dx \quad (3.15)$$

The first integral on the RHS is zero as $s_N(x)$ is a linear combination of polynomials up to $P_N^{(\alpha)}$ which are all orthogonal to $P_{N+1}^{(\alpha)}$ and hence so is $s_N(x)$. On the other hand approximating Eq. (3.15) by a quadrature gives

$$\sum_{j=0}^N u_{2N+1}(x_j) w_j = \sum_{j=0}^N s_N(x_j) P_{N+1}^{(\alpha)}(x_j) w_j + \sum_{j=0}^N r_N(x_j) w_j \quad (3.16)$$

If we choose x_j as the zeros of $P_{N+1}^{(\alpha)}$, the first sum in Eq. (3.16) vanishes and the second sum is *exactly* equal to the second integral in Eq. (3.15) which is in turn equal to the LHS of that same equation. That is to say,

$$\sum_{j=0}^N u_{2N+1}(x_j) w_j = \sum_{j=0}^N r_N(x_j) w_j = \int_{-1}^1 r_N(x) w(x) dx = \int_{-1}^1 u_{2N+1}(x) w(x) dx \quad (3.17)$$

In other words, the numerical integral of a polynomial of degree $(2N+1)$ has been exactly calculated by a $(N+1)$ -point quadrature based on Gauss nodes. Other than the

difference in quadrature points - we should say because of the difference in quadrature points - these three quadratures are different in the weights they employ. Moreover, as the quadrature points are not equispaced any more, even for one particular choice of quadrature from among the above three, we should expect the weight to vary from point to point. We have acknowledged this possibility by writing the weight as a discrete function of node location, w_j .

Before embarking on the presentation of any specific discrete Chebyshev expansion we need to address an important issue. Any discrete expansion is basically approximating the continuous expansion by using quadrature rules for the coefficients. Continuous expansion coefficients, Eq. (3.5), entail inner products and complying norms. If the multiplication of two functions (each of which is in $L_w^2[-1,1]$) lies in B_{2N} , Gauss-Radau and Gauss quadratures for the integral of this multiplication are *exact*. However, this is not necessarily true for Gauss-Lobatto quadrature as this quadrature is exact for functions in B_{2N-1} only. This is important since every continuous expansion coefficient involves a

normalization factor $\gamma_n = \left\| P_n^{(\alpha)} \right\|_{L_w^2[-1,1]}^2 = (P_n^{(\alpha)}, P_n^{(\alpha)})_{L_w^2[-1,1]}$. Discrete expansions based on

Gauss-Radau and Gauss quadratures can use the same value as that of the continuous expansion as they are exact for calculating the integral giving the norm; the integrand is the multiplication of $P_n^{(\alpha)} \in B_N$ with itself and so it lies in B_{2N} . For a discrete expansion based on Gauss-Lobatto quadrature, however, using the same normalization as the continuous case would give an inexact normalization factor for the $n = N$ term as

$\gamma_N = (P_N^{(\alpha)}, P_N^{(\alpha)})_{L_w^2[-1,1]}$ lies in B_{2N} and not B_{2N-1} . It can be shown [26] that this problem

can be obviated and for the case of Chebyshev Gauss-Lobatto quadrature the normalization factor for the $n = N$ term is π .

3.5 EQUIVALENCE OF DISCRETE POLYNOMIAL EXPANSION AND POLYNOMIAL INTERPOLATION

Now that we are clear about the discrete normalization factors needed by different Gauss quadratures, we are ready to present the *discrete polynomial expansion*, an approximation to the continuous expansion of Eq. (3.5),

$$\begin{aligned} I_N u(x) &= \sum_{n=0}^N \tilde{u}_n P_n^{(\alpha)}(x), \\ \tilde{u}_n &= \frac{1}{\tilde{\gamma}_n} \sum_{j=0}^N u(x_j) P_n^{(\alpha)}(x_j) w_j. \end{aligned} \quad (3.18)$$

When written as it is, no more can be concluded from Eq. (3.18). Let us put its pieces together; that is, substitute the expansion coefficients into the expansion and interchange the order of finite summations to obtain

$$\begin{aligned} I_N u(x) &= \sum_{n=0}^N u(x_j) \ell_j(x) \\ \ell_j(x) &= u_j \sum_{n=0}^N \frac{1}{\tilde{\gamma}_n} P_n^{(\alpha)}(x) P_n^{(\alpha)}(x_j) \end{aligned} \quad (3.19)$$

This is a staggering result as $\ell_j(x_i) = \delta_{ij}$; that is, $\ell_j(x)$ is the Lagrange interpolation polynomial based on the corresponding Gauss quadrature. Hence *the discrete polynomial expansion of a function based on Gauss quadratures interpolates it at the quadrature points!* This result provides two mathematically equivalent but computationally different means for the representation of the discrete polynomial expansion. Here, we presented it

in its general form but when we use it, we will use a particular form of it with the Lagrange interpolation polynomial based on *Chebyshev* Gauss quadratures.

3.6 DISCRETE CHEBYSHEV EXPANSION: COEFFICIENTS, NODES, AND WEIGHTS

As mentioned before, our focus will be on Chebyshev polynomials so for our purposes U will be *Chebyshev*. As discussed previously, the discrete expansion coefficients depend on the quadrature we employ and to avoid clutter we present the three Chebyshev based Gauss quadratures separately.

- Chebyshev Gauss-Lobatto *nodes, weights, and discrete expansion coefficients*

$$\begin{aligned}
 x_j &= -\cos\left(\frac{\pi}{N} j\right) & j \in [0, 1, \dots, N] \\
 w_j &= \frac{\pi}{\bar{c}_j N} & \bar{c}_j = \begin{cases} 2, j = 0, N \\ 1, j \in [1, 2, \dots, N-1] \end{cases} \\
 \tilde{u}_n &= \frac{2}{\bar{c}_n N} \sum_{j=0}^N \frac{1}{\bar{c}_j} u(x_j) T_n(x_j)
 \end{aligned} \tag{3.20}$$

- Chebyshev Gauss-Radau *nodes, weights, and discrete expansion coefficients*

$$\begin{aligned}
 x_j &= -\cos\left(\frac{2\pi}{2N+1} j\right) & j \in [0, 1, \dots, N] \\
 w_j &= \frac{\pi}{c_j (2N+1)} & c_j = \begin{cases} 2, j = 0 \\ 1, j \in [1, 2, \dots, N] \end{cases} \\
 \tilde{u}_n &= \frac{2}{c_n (2N+1)} \sum_{j=0}^N \frac{1}{c_j} u(x_j) T_n(x_j)
 \end{aligned} \tag{3.21}$$

- Chebyshev Gauss *nodes, weights, and discrete expansion coefficients*

$$\begin{aligned}
 x_j &= -\cos\left(\frac{(2j+1)\pi}{2N+2}\right) & j \in [0, 1, \dots, N] \\
 w_j &= \frac{\pi}{N+1} & \text{for all } j \in [0, 1, \dots, N] \\
 \tilde{u}_n &= \frac{2}{c_n(N+1)} \sum_{j=0}^N u(x_j) T_n(x_j)
 \end{aligned} \tag{3.22}$$

To get a feel for the different node distributions given by the above equations Figure 3-2 can be examined.

Figure 3-2 Chebyshev Gauss-Lobatto, Chebyshev Gauss-Radau, and Chebyshev Gauss quadrature nodes. Note that $x = 1$ for Chebyshev Gauss-Radau and $x = \pm 1$ for Chebyshev Gauss are *not* among the quadrature points.

As mentioned before, one reason for taking up Chebyshev polynomials as the basis functions is the availability of such closed-form expressions as those just given. This argument, however, must not be taken too strongly as late Golub has shown that with the three-term recurrence relation of any orthogonal polynomial available, the associated quadrature rule can be generated by calculating the eigenvalues and the first component of the orthonormalized eigenvector of a symmetric tridiagonal matrix [27]. He also presents an algorithm to actually compute that three-term recurrence relation based on the

moments of the weight function $\mu_k = \int_a^b w(x)x^k dx, k = 0, 1, \dots, 2N$.

3.7 NON-PERIODIC DIFFERENTIATION TOOLS: MATRICES AND EXPANSION COEFFICIENTS

We are now at a place to give the crux of the matter. We are to approximate the derivative of a function using the discrete Chebyshev expansion. From one point of view, we can work with the expansion coefficients and from another with the Lagrange interpolation polynomial. We have already given the results for the former but for the *continuous* Chebyshev expansion, Eq. (3.13). If there is any justice, substituting the discrete coefficients for continuous ones in that equation should give us the discrete analogue, and it does. That would have worked for any other polynomial expansion too, had we have the recurrence relation for its corresponding derivative operator. For the special case of the discrete Chebyshev expansion, because of the nature of the basis, $\cos(n\theta)$, we can go further; we can use fast Fourier transform (FFT) techniques to calculate the expansion coefficients. As an example, the discrete Chebyshev expansion based on Chebyshev Gauss-Lobatto quadrature, using Eq. (3.20) with

$$T_n(x) = \cos(n \cos^{-1}(x)) = \cos(nj\pi / N) \text{ yields } \tilde{u}_n = \frac{2}{\bar{c}_j N} \sum_{j=0}^N \frac{1}{c_j} u(x_j) \cos(nj\pi / N) \text{ which is}$$

a discrete cosine expansion and can be evaluated using FFT.

An alternative approach is based on using the Lagrange interpolation polynomial, Eq. (3.19). The derivative of a function can be approximated by differentiating the global polynomial $\ell_j(x)$ since

$$I_N \frac{d}{dx} I_N u(x_i) = \sum_{j=0}^N u(x_j) \frac{d}{dx} \ell_j(x) |_{x_i} = \sum_{j=0}^N D_{ij} u(x_i), \quad (3.23)$$

in which D_{ij} are elements of the $(N+1) \times (N+1)$ differentiation matrix based on the corresponding Lagrange interpolation polynomial.

It is informative to give explicit expressions for the Lagrange interpolation polynomial and its corresponding differentiation matrix for the symmetric Chebyshev Gauss quadratures.

- Chebyshev Gauss-Lobatto *Lagrange interpolation polynomial and Differentiation Matrix*

$$\ell_j(x) = \frac{(-1)^{N+j+1}(1-x^2)T'_N(x)}{\bar{c}_j N^2(x-x_j)} \quad (3.24)$$

$$D_{ij} = \begin{cases} -(2N^2+1)/6 & i=j=0, \\ (2N^2+1)/6 & i=j=N, \\ \frac{-x_i}{2(1-x_i^2)} & i=j \in [1, \dots, N-1], \\ \frac{\bar{c}_i}{\bar{c}_j} \frac{(-1)^{i+j}}{(x_i-x_j)} & i \neq j. \end{cases} \quad (3.25)$$

- Chebyshev Gauss *Lagrange interpolation polynomial and Differentiation Matrix*

$$\ell_j(x) = \frac{T_{N+1}(x)}{(x-x_j)T'_{N+1}(x_j)} \quad (3.26)$$

$$D_{ij} = \begin{cases} \frac{x_i}{2(1-x_i^2)} & i=j, \\ \frac{T'_{N+1}(x_i)}{(x_i-x_j)T'_{N+1}(x_j)} & i \neq j. \end{cases} \quad (3.27)$$

An interesting property of a differentiation matrix based on Gauss-Lobatto or Gauss quadratures is its being centro-antisymmetric,

$$D_{ij} = -D_{N-i, N-j} \quad (3.28)$$

which follows from the even-odd symmetry of the ultraspherical polynomials and hence the symmetry of the quadrature nodes around $x = 0$. This kinship is shown, via using the same color for corresponding elements, in Table 3-1 for a 6-by-6 Chebyshev Gauss-Lobatto differentiation matrix.

-8.5	10.47214	-2.89443	1.52786	-1.10557	0.5
-2.61803	1.17082	2	-0.89443	0.61803	-0.27639
0.72361	-2	0.17082	1.61803	-0.89443	0.38197
-0.38197	0.89443	-1.61803	-0.17082	2	-0.72361
0.27639	-0.61803	0.89443	-2	-1.17082	2.61803
-0.5	1.10557	-1.52786	2.89443	-10.47214	8.5

Table 3-1 6-by-6 Chebyshev Gauss-Lobatto differentiation matrix and its centro-antisymmetry

It is interesting to look at the Lagrange interpolation polynomial from a view point different than the one given in Eq. (3.19). Let us examine the U Gauss case as an example. For this case the interpolation points were the zeros of $\phi_{N+1}(x)$. An interpolation polynomial $\ell_j(x)$ is zero everywhere except at its support point x_j . Thus $\phi_{N+1}(x)$ is quite close to $\ell_j(x)$ that we are looking for; we just have to define a new value for it at the support point. Taylor expanding near that point we have

$$\phi_{N+1}(x) \approx \phi_{N+1}(x_j) + \phi'_{N+1}(x_j)(x - x_j) + O((x - x_j)^2) \quad (3.29)$$

$\phi_{N+1}(x_j)$ is zero since we are using U Gauss points hence if we eliminate the root-at- x_j factor $(x - x_j)$ in $\phi_{N+1}(x)$ and normalize its value there we are left with our U Gauss based Lagrange interpolation polynomial which in our particular case of U being Chebyshev (and so $\phi_{N+1}(x)$ being $T_{N+1}(x)$) is exactly Eq. (3.26) [28].

Figure 3-3 illustrates, as an example, what a Chebyshev Gauss-Lobatto differentiation matrix looks like, for $N = 15$; that is, it displays a 16-by-16 matrix corresponding to a 16 point quadrature grid. Notice that large elements are in the first and last rows, corresponding to differentiation near the boundaries $x = -1$ and $x = 1$, respectively. According to Eq.(3.25), the first element, in (1,1) position, is $-(2 \times 15^2 + 1)/2 = -75.1666$, but that is not the largest element. The largest element is in (1,2) position equaling 91.5231. We already know that there are elements as small as 0.5 in *every* Chebyshev Gauss-Lobatto matrix but they are not the smallest here. The smallest element (in magnitude) is in (8,8) position equaling 0.0528 and that makes this matrix *dense*, as opposed to the case of finite differences which have sparse differentiation matrices.

Figure 3-4 shows the degree 15 Lagrange interpolation polynomials $\ell_4(x)$ and $\ell_7(x)$ based on the Chebyshev Gauss-Lobatto quadrature nodes and supported at x_4 and x_7 , respectively. This is polynomials that leads to the differentiation matrix illustrated in Figure 3-3. Note that, like any other cardinal function, $\ell_4(x)$ is zero at every quadrature point except for the one it is supported at. Note, however, that unlike the trigonometric cardinal function (periodic sinc function) of the previous chapter, polynomial cardinal functions are *not* identical in shape. Specifically, it is seen that $\ell_7(x)$ is *not* obtained by shifting $\ell_4(x)$ three nodes to the right. This is explicitly indicated on the figure by showing the *different* values of the minimums occurring right after the peaks.

Figure 3-3 Chebyshev Gauss-Lobatto differentiation matrix, D_{ij} of Eq. (3.25) with $N = 15$.

Figure 3-4 Degree 15 Lagrange interpolation polynomials $\ell_4(x)$ and $\ell_7(x)$ based on Chebyshev Gauss-Lobatto nodes and supported at x_4 and x_7 , respectively.

Higher order derivatives can be obtained from the powers of the first order differentiation matrix, as the following holds for all U Gauss based methods

$$D^{(q)} = (D)^q \quad (3.30)$$

In [29] it is shown that $D^{(q)}$ can be obtained from $D^{(q-1)}$ with only five flops per element and a necessary and sufficient condition for Eq. (3.30) to hold is presented for polynomial and Hermite expansions.

3.8 AMELIORATING ROUND OFF EFFECTS IN POLYNOMIAL METHODS

Finite precision effects inevitably affect the approximation results of any method and polynomial methods are no exception to this rule. Here we discuss the effect of roundoff on the eigenvalues of U Gauss based differentiation matrices and on derivative approximations.

Theoretically, the differentiation matrix based on polynomial methods is nilpotent since $N+1$ times differentiating a polynomial of degree N yields an identically zero result and since the interpolating polynomial of degree N is exact for polynomials of degree N , so is the differentiation matrix D [20]. Nilpotency of D is equivalent to its having $N+1$ zero eigenvalues but only a one dimensional eigenspace; D is similar to a single Jordan block of dimension $N+1$. In other words, $D[1,1,\dots,1]^T = [0,0,\dots,0]^T$, and $[1,1,\dots,1]^T$ is the only eigenvector. A corollary of this argument is that the elements in every row of D must sum to zero, as the derivative of a constant function is zero at every point of the quadrature grid. In consequence, we have

$$D_{ii} = -\sum_{\substack{j=0 \\ j \neq i}}^N D_{ij} \quad (3.31)$$

If we calculate the eigenvalues of D on a double-precision machine, however, we encounter an apparently contradictory result; as seen in Figure 3-5, the eigenvalues are *not* zero. The reason is that $N+1$ zero eigenvalues correspond to the characteristic

polynomial z^{N+1} . As a result, eigenvalues will move distances of order $\varepsilon^{1/(N+1)}$ when a error of size ε is present in the matrix elements [30]. They approximately lie on a circle of radius 1.2. The condition numbers of these eigenvalues are all of order 10^{13} , signaling the presence of repeated eigenvalues.

Figure 3-5 Double-precision calculated eigenvalues of 16-by-16 Chebyshev Gauss-Lobatto differentiation matrix. In exact arithmetic they are all zero.

Figure 3-6 illustrates the minimum and maximum magnitudes of double-precision-calculated eigenvalues of 10-by-10 through 200-by-200 Chebyshev Gauss-Lobatto differentiation matrices. We have increased the size of test matrices in steps of 10. It is seen that the, in double precision, the minimum magnitude of the eigenvalues has no special trend and behaves rather randomly.

The maximum magnitudes, on the other hand, increase monotonically and they approximately obey the quadratic $0.018(N+1)^2 - 0.4(N+1) + 5.8$.

Figure 3-6 Minimum and maximum magnitudes of double-precision-calculated eigenvalues of 10-by-10 through 200-by-200 Chebyshev Gauss-Lobatto differentiation matrices (size increase in steps of 10).

Coming to the effect of roundoff on derivative approximation, we note that if Eq. (3.25) is used, per se, to generate the Chebyshev Gauss-Lobatto differentiation matrix we encounter another seemingly contradictory phenomenon; the L^∞ error would grow as N^4 for the first derivative and N^6 for the second derivative [31]. This is surprising at first as error analysis for matrix-vector multiplication suggests an accumulation error of order N^2 [32]. Furthermore, the error in the second derivative stays of the same order whether we use $D^{(2)} = D^2$ or the actual $D^{(2)}$ with the elements derived from scratch. This signals that the culprit cannot be accumulation and we should investigate the computation of the

matrix elements themselves. As shown in Figure 3-3, and more clearly the logarithmic gray scale of Figure 3-7, the greatest elements of D are aggregated at the top-left and bottom-right corners of it; hence any error in computing these elements will have a large effect on the differentiation results. As noted earlier, the maximum element of D lies in (1,2) position and, from Eq. (3.25), (the $i \neq j$ part), it involves the calculation of the first and second quadrature points. The first quadrature point in U Gauss-Lobatto quadratures is always $x_0 = -1$ and so that one is known *and* can be stored exactly in finite precision. The second quadrature point is $x_1 = \cos(\pi / N)$ which cannot be stored exactly in finite precision. If a roundoff of δ is incurred in its computation, then, for large N , x_1 may be expanded in the Taylor series

$$x_1 = 1 - \frac{\pi}{2N^2} + \delta + O(N^{-4}, \delta^2) \quad (3.32)$$

and in turn the maximum element D_{01} can be expanded as

$$D_{01} = \frac{-2}{x_0 - x_1} \approx \frac{-2}{\delta + \pi / (2N^2)} = -4N^2 + O(N^4 \delta) \quad (3.33)$$

It is now seen that the order N^4 error in approximating the first derivative by matrix multiplication is caused by roundoff and it dominates the accumulation error of order N^2 . Notice that this dramatic roundoff error is caused by the catastrophic cancellation in the denominator of Eq.(3.33); as seen in Figure 3-2, the first and second nodes of Chebyshev Gauss-Lobatto quadrature are very close to each other and for large N they almost coincide.

Figure 3-7 Logarithmic gray scale image of absolute value of elements of a 128-by-128 Chebyshev Gauss-Lobatto differentiation matrix. Lighter means larger as seen from the logarithmic color bar on the right.

This error could have been expected by scrutinizing Eq. (3.31) too; in double precision

the equations $D_{00} = -\sum_{\substack{j=0 \\ j \neq 0}}^N D_{0j}$ and $D_{NN} = -\sum_{\substack{j=0 \\ j \neq N}}^N D_{Nj}$ do not hold. D_{00} and D_{NN} are

calculated and stored exactly but the terms in the RHS sums – particularly those right next to the corners – are subject to roundoff due to the catastrophic cancellation mentioned before.

The first idea to obviate this roundoff artifact may be to pre-calculate the matrix elements; this works but is rather inconvenient as it is not amenable to automatic implementation for different numbers of quadrature points. A successful alleviation comes along by *making* Eq. (3.31) hold [33]. To that end, we use Eq. (3.25) for the off-

diagonal elements of D and then obtain the diagonal entries from Eq. (3.31). Another method proposed [34] is to use trigonometric identities to arrive at

$$D_{ij} = \begin{cases} -(2N^2 + 1)/6 & i = j = 0, \\ (2N^2 + 1)/6 & i = j = N, \\ \frac{-x_i}{2 \sin^2(i\pi/N)} & i = j \in [1, \dots, N-1], \\ -\frac{\bar{c}_i}{2\bar{c}_j} \frac{(-1)^{i+j}}{\sin((i+j)\pi/2N) \sin((i-j)\pi/2N)} & i \neq j. \end{cases} \quad (3.34)$$

Using this matrix gains an order of convergence in L^∞ error. If we exploit the centro-antisymmetry of the Chebyshev Gauss-Lobatto or Chebyshev Gauss differentiation matrices and also employ the initialization just given, the growth of L^∞ will be reduced by one order again and we will be left with an N^2 rate for the first derivative.

Another resolution is based on using the change of independent variable $x = \arcsin(\alpha y) / \arcsin(\alpha)$ which leads to a better conditioned differentiation matrix. [35].

The idea is that the independent variable is changed, through the stretching function on the right hand side, to one that is not so densely clustered near the ends. This is logical as we see in Eq. (3.32) that the too-densely-spaced points near the boundaries are to blame for the ill-conditioning of the Chebyshev Gauss Lobatto differentiation matrix and unless so sharp gradients are present at the boundaries that would really justify the need for a closely packed grid, the only reason why one has to employ such a high density clustering is because otherwise the underlying *polynomial* interpolation threatens them by Runge phenomenon. A standard polynomial based method in the y variable as defined

above, however, corresponds to a *non-polynomial* method in the x variable, with basis functions $T_j\left(\frac{\sin(\arcsin \alpha x)}{\alpha}\right)$, and solves the problem.

3.9 POLYNOMIAL INTERPOLATION: LAGRANGE, NEWTON, OR BARYCENTRIC FORM? THAT IS THE QUESTION

In section 3.5 we noticed that a reordering of the discrete polynomial expansion (discretized based on approximating the continuous coefficients by Gauss quadratures) gives us the equivalent polynomial interpolation. Specifically, we arrived at the Lagrange form for polynomial interpolation and in section 3.7 we actually presented the Lagrange cardinal function $\ell_j(x)$ for some node distributions. However, Lagrange form, although of analytical utility and beauty, is deprecated in practice [36]. Let us first see the shortcomings of Lagrange form as a motivation for a quest for the right form for numerical implementations.

If the vector space of all polynomials of degree N or smaller is denoted by $\prod_n$ the problem of finding a polynomial $p \in \prod_n$ interpolating u at the knots x_j has a unique solution continuously dependent on the data and is hence well-posed. We have already seen that the Lagrange form of this polynomial interpolant is

$$\begin{aligned} p(x) &= \sum_{j=0}^N \ell_j(x) u_j, \\ \ell_j(x) &= \frac{\prod_{i=0, i \neq j}^N (x - x_i)}{\prod_{i=0, i \neq j}^N (x_j - x_i)}. \end{aligned} \tag{3.35}$$

It is seen that each evaluation of $p(x)$ requires $O(N^2)$ additions and multiplications and each new pair of data forces us to do all that from scratch. To add insult to injury, the computation is unstable!

Newton's approach ameliorates the burden of $O(N^2)$ operations by introducing a recursive formula that requires around $N^2 / 2$ divisions and N^2 subtractions to create the divided differences

$$u[x_j, x_{j+1}, \dots, x_{i-1}, x_i] = \frac{u[x_{j+1}, \dots, x_i] - u[x_j, \dots, x_{i-1}]}{x_i - x_j}, \quad (3.36)$$

with the initial condition $f[x_j] = f_j$. It then generates the interpolant by the *Newton interpolation formula* which needs only N flops

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1, \dots, x_N](x - x_0)(x - x_1) \dots (x - x_{N-1}). \quad (3.37)$$

Lagrange formula can be improved by noting that the *numerator* of $\ell_j(x)$ can be written as the multiplication of *all* difference terms $(x - x_i)$ divided by the one we should dispense with, $(x - x_j)$. To wit, $\overbrace{(x - x_0)(x - x_1) \dots (x - x_N)}^{\ell(x)} / (x - x_j) = \ell(x) / (x - x_j)$. With the barycentric weights defined by

$$w_j = \frac{1}{\prod_{i \neq j} (x_j - x_i)} = \frac{1}{\ell'(x_j)}, \quad (3.38)$$

which gives the polynomial cardinal function in the form

$$\ell_j(x) = \ell(x) \frac{w_j}{x - x_j} \quad (3.39)$$

and that, after substitution in Eq. (3.35), in turn yields the *improved Lagrange formula*

$$p(x) = \ell(x) \sum_{j=0}^N \frac{w_j}{x - x_j} u_j, \quad (3.40)$$

which needs $O(N^2)$ flops to compute the weights and then only $O(N)$ flops to evaluate the interpolant.

We are one step away from the barycentric formula we set out to find. Observing that for data points all equal to one, the polynomial interpolant is also equal to one, $p(x) = 1$, and using Eq.(3.40) and Eq. (3.35) we have

$$1 = \sum_{j=0}^N \ell_j(x) = \ell(x) \sum_{j=0}^N \frac{w_j}{x - x_j}. \quad (3.41)$$

Since it is 1, Eq. (3.41) can be used in Eq. (3.40) to kill off the $\ell(x)$ factor and leave us with a pleasingly symmetric formula which is none but the *barycentric polynomial interpolation formula*

$$p(x) = \frac{\sum_{j=0}^N \frac{w_j}{x - x_j} u_j}{\sum_{j=0}^N \frac{w_j}{x - x_j}} \quad (3.42)$$

Using Eq. (3.38) and the explicit polynomial cardinal functions derived for different Chebyshev Gauss points, we can get explicit formulas for the corresponding interpolation weights. For Chebyshev Gauss points the weights become

$$w_j = (-1)^j \sin \frac{(2j+1)\pi}{(2N+2)} \quad (3.43)$$

and for Chebyshev Gauss-Lobatto point they are

$$w_j = (-1)^j / \tilde{c}_j. \quad (3.44)$$

Note that as the barycentric formula is in the form of a division with weights appearing in exactly the same way in the numerator and denominator, the common factors in the weights can be dispensed with. For instance, linear transformation of $[-1,1]$ to $[a,b]$ multiplies the weights in Eq. (3.44) by $2^N(b-a)^{-N}$ but we need not include this due to the reasoning just mentioned.

Higham has rigorously shown that the improved Lagrange formula, Eq.(3.40), is backward stable while the barycentric formula, Eq.(3.42), is not. The latter is only forward stable for any set of interpolation nodes with a “small” Lebesgue constant [37]. However, due to the practical advantage of scalability of weights in the latter, it must become the method of choice.

With only a handful of points one might doubt the true superiority of the barycentric formula over Lagrange formula as the two yield the same result as illustrated in Figure 3-8. We have chosen a function with an absolute value function in it to create a kink at the origin. When we go to a higher number of points, to be able to resolve the kink, the curtain falls. The Lagrange formula, as is clearly seen in Figure 3-9, results in a badly-conditioned polynomial that blows up. Barycentric formula, on the other hand, has no problem dealing with the increased number of nodes.

This section might seem like a digression from spectral methods but it is not! Barycentric representation of polynomial interpolants gives us yet another way of calculating the differentiation matrices. What is more, it turns out that this is by far the least prone to roundoff errors. From Eq. (3.42), we see that the barycentric representation of the cardinal function $\ell_j(x)$ is

$$\ell_j(x) = \frac{\frac{w_j}{x - x_j}}{\sum_{j=0}^N \frac{w_j}{x - x_j}}, \quad (3.45)$$

Figure 3-8 Polynomial interpolation approximating $|x| + x - x^7$ on an 11-point Chebyshev Gauss-Lobatto grid. *Top:* Using barycentric formula. *Bottom:* Using Lagrange formula.

which may be written as

$$\ell_j(x) \sum_{j=0}^N \frac{w_j}{x - x_j} (x - x_i) = \frac{w_j}{x - x_j} (x - x_i). \quad (3.46)$$

Differentiating with respect to x , this yields

$$\ell'_j(x) + \ell_j(x) \left(\sum_{j=0}^N \frac{w_j (x - x_i)}{x - x_j} \right)' = w_j \left(\frac{x - x_i}{x - x_j} \right)' \quad (3.47)$$

therefore

$$\ell'_j(x_i) = \frac{w_j / w_i}{x_i - x_j} \quad (3.48)$$

Figure 3-9 Polynomial interpolation approximating $|x| + x - x^7$ on a 101-point Chebyshev Gauss-Lobatto grid. *Top*: Using barycentric formula and full kink resolution. *Bottom*: Using Lagrange formula which results in a badly-conditioned polynomial interpolant.

Notice the ease with which we reached Eq. (3.48). That comes from the symmetry features of the barycentric formula.

3.10 LEBESGUE FUNCTION AND QUALITY OF INTERPOLANTS

The rationale we gave in section 3.4 for choosing the quadrature points based on the zeros or extrema of ultraspherical polynomials U was that we wanted to do better: *any* $N+1$ points were able to give an exact quadrature for polynomials of order up to N but with the points carefully chosen we could get exact quadratures for polynomials of order

up to $2N+1$. That reasoning was motivational but not sufficient enough to put out every argument thrown at it. One could argue that if they did not need that “extra exactness” – say if they were not as greedy and were comfortable with providing $N+1$ data pairs to obtain exact quadratures for polynomials *up to* order N – then they could go ahead and choose an *arbitrary* set of quadrature points. Indeed, they could, but only if they were not to use the results of the quadratures, namely the discrete coefficients, to approximate the continuous coefficients of a polynomial expansion on that *arbitrary* grid. In other words, only if they were not to interpolate on that arbitrary grid would they be safe. Let us see why this is, qualitatively at first.

Figure 3-10 Comparison of polynomial interpolation based on equispaced and Chebyshev Gauss-Lobatto nodes.

Figure 3-10 illustrates interpolation based on 10 data pairs sampled from $u(x)=1/(1+x^2)$ (Top panes) and $u(x)=1/(1+16x^2)$ (Bottom panes). In the left panes the samples are equispaced and so the polynomial interpolation is based on an equispaced grid. The right panes show the results for samples taken at Chebyshev Gauss-Lobatto points. Notice that the polynomial interpolant to $u(x)=1/(1+16x^2)$ based on equispaced samples blows up near the ends. This is the Runge phenomenon. Contrariwise, the polynomial interpolant to $u(x)=1/(1+x^2)$ does not show any divergence behavior, in spite of interpolation samples being taken at equispaced points again. This qualitatively suggests that the success of polynomial interpolation should depend on at least two factors: the particular function being interpolated (i.e., where the samples come from) and the particular spacing of the nodes (i.e., how the samples are taken). It is not always easy to determine the quality of the interpolation qualitatively; we need to be able to measure that quantitatively.

The operator that maps a function to its polynomial interpolant is *linear* and we have seen that on U Gauss grids this interpolation operator is the same as the discrete projection operator. As we will be dealing with the same grid points we do not change the notation $I_N u = u_N$. It turns out the norm of this operator can provide a good measure of the behavior of the interpolant

$$\Lambda_N = \|I_N\| = \sup_{u \in C^0[-1,1]} \frac{\|I_N u\|}{\|u\|} \quad (3.49)$$

$$\|u\| = \max_{x \in [-1,1]} |u(x)|$$

Λ_N is known as the *Lebesgue constant*. If we define $\lambda_N(x) = \sum_{j=0}^N |\ell_j(x)|$ as the Lebesgue function, the Lebesgue constant becomes $\Lambda_N = \max_{x \in [-1,1]} \lambda_N(x)$. Then it can be seen [38] that the Lebesgue constant is a measure of how closely a given polynomial interpolant of degree N , $I_N u(x)$, is able to approximate a $C^0[-1,1]$ function $u(x)$, compared with the optimal interpolant of the same (or lower) degree, $p^*(x)$,

$$\|u - I_N u\|_\infty \leq (1 + \Lambda_N) \|u - p^*\|_\infty \quad (3.50)$$

The Lebesgue constant for *any* polynomial interpolant, and for all values of N , is bounded as

$$\Lambda_N \geq \frac{2}{\pi} \ln(N+1) + A \quad (3.51)$$

where $\gamma = 0.577221566$ is Euler's constant and in the limit of large N , $A = (2/\pi)(\gamma + \ln(4/\pi))$. This indicates that even in the optimal case, with increasing number of grid points, the maximum error in interpolation grows as $\log(N)$. This rules out the existence of a set of grid points with a constant Lebesgue constant. Moreover, and this is even worse, Lebesgue constant depends, in a sensitive way, on the distribution of the interpolation nodes. For equispaced polynomial interpolation the situation is far from optimal,

$$\frac{2^{N-2}}{N^2} \leq \Lambda_N^{\text{equispaced}} \leq \frac{2^{N+3}}{N}, \quad (3.52)$$

which in the limit becomes [39],

$$\Lambda_N^{\text{eq}} = \frac{2^{N+1}}{eN(\ln N + \gamma)} \text{ as } N \rightarrow \infty \quad (3.53)$$

This divergence is too rapid to leave any chance for this interpolant to be used in practice; equispaced polynomial interpolation is ill-posed.

For polynomial interpolation on Chebyshev Gauss-Lobatto points, the Lebesgue constant is exceptionally close to the theoretical optimum [40] and is even superior to that of the Chebyshev Gauss points,

$$\begin{aligned}\Lambda_N^{\text{Chebyshev-Gauss}} &\leq \frac{2}{\pi} \ln(N+1) + A + \frac{2}{\pi} \ln(2), \\ \Lambda_N^{\text{Chebyshev Gauss-Lobatto}} &\leq \Lambda_{N-1}^{\text{Chebyshev-Gauss}}.\end{aligned}\tag{3.54}$$

There must be something special about Chebyshev Gauss-based points that makes their Lebesgue constant near optimal. Notice in Figure 3-10 that, when we use equispaced polynomial interpolation, the largest errors happen near the boundaries. It is then logical to space the grid sparsely in the middle where we have high accuracy any way and make it denser near the end points. That is exactly what is special about Chebyshev Gauss based points. They are clustered more densely near the ends than the center of the interval. In fact, this property is not exclusive to Chebyshev Gauss based grids and is shared by all of the U Gauss based grids; that is, *zeros of all ultraspherical polynomials are clustered densely near the ends of the interval*. They all share a minimum grid spacing of the form

$$\Delta x_{\min} = 1 - cN^2\tag{3.55}$$

where c depends on the particular U used to generate the grid.

As an example, the minimum grid size for Chebyshev Gauss-Lobatto points is

$$\begin{aligned}\Delta x_{\min} &= -\cos\left(\frac{\pi}{N}\right) - (-1) \\ &\approx \frac{\pi^2}{2} \frac{1}{N^2}, \quad N \gg 1\end{aligned}\tag{3.56}$$

This implies quadratic clustering of the nodes near the ends. On the other hand, away from the ends and close to the center the spacing is a maximum

$$\begin{aligned}\Delta x_{\max} &= -\cos\left(\frac{N}{2}\frac{\pi}{N}\right) + \cos\left(\left(\frac{N}{2}-1\right)\frac{\pi}{N}\right) = \sin\left(\frac{\pi}{N}\right) \\ &\approx \pi \frac{1}{N}, \quad N \gg 1\end{aligned}\tag{3.57}$$

This implies linear spacing away from the boundaries.

It is now clear that the only structural condition that must hold for the grid points in high-order polynomial interpolation is quadratic clustering near the ends. This has the important implication that otherwise the location of the knots does *not* matter in polynomial interpolation. As a case in point, the bound on the Lebesgue constant for Legendre Gauss-Lobatto points is

$$\Lambda_N^{\text{Legendre Gauss-Lobatto}} \leq \frac{2}{\pi} \ln(N+1) + 0.685\dots,\tag{3.58}$$

which is of the same order as that of Chebyshev Gauss-Lobatto points. Thus, there is nothing that stops us from searching for node distributions that are even closer to the theoretical optimal, as measured by the Lebesgue constant. One step closer to the optimum are the grid points

$$x_j = -\frac{\cos\left(\frac{2j+1}{2N+2}\pi\right)}{\cos\left(\frac{\pi}{2N+2}\right)}\tag{3.59}$$

known as the extended Chebyshev Gauss grid points. Their associated Lebesgue constant is

$$\Lambda_N^{\text{extended Chebyshev Gauss}} = \frac{2}{\pi} \ln(N+1) + A + \frac{2}{\pi} \left(\ln 2 - \frac{2}{3} \right)\tag{3.60}$$

which is even closer to the optimal than that of the Chebyshev Gauss-Lobatto points.

To explore node clustering, it is customary to consider a *node density function* $\mu(x)$ to describe the variation of the density of nodes over the interval of interest. We normalize this function according to,

$$\int_{-1}^1 \mu(x) dx = 1 \quad (3.61)$$

With this normalization, each interval of length dx will have $N\mu(x)dx$ nodes, giving a total number of $\int_{-1}^1 N\mu(x)dx$ nodes, which equals N . This yields the approximate distance

$1/N\mu(x)$ between two consecutive nodes.

In the case of equispaced interpolation, the node density function is constant and Eq. (3.61) gives,

$$\begin{aligned} \int_{-1}^1 \mu dx &= \mu \int_{-1}^1 dx = 1 \\ \mu_{\text{equispaced}} &= 1/2 \end{aligned} \quad (3.62)$$

All Jacobi polynomials obey the node density function (per unit length),

$$\mu(x) = \frac{1}{\pi\sqrt{1-x^2}} \quad (3.63)$$

Node density functions are not only interesting in their own right but are also the core of potential functions associated with node distributions.

3.11 POTENTIAL FUNCTION AND CONVERGENCE RATE OF INTERPOLANTS

We explained the superiority of a quadratically clustered polynomial interpolation grid to an equispaced one using the Lebesgue constants associated with them. Here take a step

further and appeal to the potential theory to scrutinize what is going on under the hood, exactly.

We remember that a *Taylor series is convergent in the largest singularity-free circle around the expansion point*. We may generalize this result to the case of polynomial interpolants where, instead of being aggregated all at one point, the nodes are spread over an interval. Related to the node density function of an interpolation is a potential function

$$\phi(z) = -\int_{-1}^1 \mu(x) \ln |z - x| dx + C \quad (3.64)$$

We could take the potential function of Eq. (3.64) as given but it is informative to delve in and see its source. As we are to identify the grid point distributions that lead to well-behaved interpolations, we are seeking the smallest interpolation errors possible. The Cauchy remainder for interpolation is

$$R_N(z) = u(z) - I_N u(z) = \frac{u^{(N+1)}(\xi)}{(N+1)!} \ell(z) \quad (3.65)$$

Therefore,

$$|\ell(z)| = e^{\sum_{j=0}^N \ln |z - x_j|} \quad (3.66)$$

Now if we consider $N+1$ electrostatic charges each placed at a node x_j , we can define a potential function associated with them which for large values of N becomes

$$\phi(z) = \lim_{N \rightarrow \infty} \frac{1}{N+1} \sum_{j=0}^N \ln |z - x_j| = \int_{-1}^1 \mu(x) \ln |z - x| dx. \quad (3.67)$$

This yields,

$$\lim_{N \rightarrow \infty} |\ell(z)|^{1/N} = e^{-\phi(z)}. \quad (3.68)$$

The $\frac{u^{(N+1)}(\xi)}{(N+1)!}$ part of the remainder entails the function being interpolated and in the limit of large N it converges to $e^{\phi(z_0)}$, where z_0 is the limiting singular point of u . Putting these two contributions together we obtain

$$\alpha(z) = \lim_{N \rightarrow \infty} |R(z)|^{1/N} = e^{\phi(z_0) - \phi(z)}. \quad (3.69)$$

It turns out that the equivalents of singularity-free circles for polynomial interpolation are equi-potential curves given by Eq. (3.64). Specifically, a polynomial $I_N u(z)$ interpolating an analytic function $u(z)$ (at $N+1$ nodes on the interval $[-1, 1]$) converges to $u(z)$ inside the largest *singularity-free equi-potential curve of $\phi(z)$* . We have divergence outside that curve and when it happens it is dubbed Runge phenomenon. If z_0 denotes the singular point of $u(z)$ lying ‘just outside’ that largest singularity-free equi-potential curve, the rate of convergence or divergence would be exponential, like $\alpha^N(z)$ where $\alpha(z) = e^{\phi(z_0) - \phi(z)}$ [20].

For an equispaced interpolation, integrating the potential Eq. (3.64) using the corresponding node density function, Eq. (3.62), gives,

$$\phi^{\text{equispaced}}(z) = -\frac{1}{2} \text{Re}[(1-z) \ln(1-z) - (-1-z) \ln(-1-z)] + C. \quad (3.70)$$

With the particular values $\phi^{\text{equispaced}}(0) = C$ and $\phi^{\text{equispaced}}(\pm 1) = -\ln(2) + C$. This gives $\alpha(\pm 1) = e^{\phi(0) - \phi(\pm 1)} \alpha(0) = 2\alpha(0)$; i.e., an equispaced polynomial interpolant on $[-1, 1]$ takes values 2^N times larger at the ends ($x = \pm 1$) than at the center ($x = 0$). Note that this *does not* necessarily mean we will have divergence at the boundaries unless the boundaries

happen to fall in an equi-potential curve in which a singularity of $u(z)$ dwells since we have $\alpha(\pm 1) = e^{\phi(z_0) - \phi(\pm 1)} = e^{\phi(z_0)} e^{\ln 2} = 2e^{\phi(z_0)}$; that is, only if $e^{\phi(z_0)}$ falls below $1/2$, does divergence occur at the ends. This, however, means that even if we have convergence near the boundaries, it is painfully slow, 2^N times slower than that at the center.

For Jacobi polynomials we obtain

$$\phi(z)^{\text{Jacobi}} = -\ln \left| z + \sqrt{z^2 - 1} \right| + C. \quad (3.71)$$

These are confocal ellipses with foci ± 1 and are called Bernstein regularity ellipses [41]. Some algebraic manipulation of Eq. (3.71) yields the explicit equation of these ellipses [20]

$$\frac{x^2}{((\alpha + 1/\alpha)/2)^2} + \frac{y^2}{((\alpha - 1/\alpha)/2)^2} = 1 \quad (3.72)$$

Of utmost importance to us is a degenerate ellipse, the ridge from -1 to 1; that is exactly the $[-1, 1]$ interval. Everywhere along that ridge the potential is constant; $\phi(z)^{\text{Jacobi}}|_{z \in [-1, 1]} = C$. That is the maximum value of this potential too. The existence of this degenerate ellipse and the fact that it still is an equi-potential curve implies that convergence on $[-1, 1]$ does *not* need homomorphism anywhere outside that interval. In fact, even with a singularity on that ridge we would not necessarily have divergence of interpolation at a point there as $\alpha(z)|_{z \in [-1, 1]} = e^{\phi(z_0) - \phi(z)^{\text{Jacobi}}|_{z \in [-1, 1]}} = e^{\phi(z_0)}$; no divergence occurs unless $e^{\phi(z_0)}$ falls below 1.

Figure 3-11 clearly depicts the difference between the equi-potential curves associated with equispaced polynomial interpolation and those associated with polynomial interpolation based on quadratically clustered grids. Note that the former curves resemble

American footballs of increasing size while the latter are ellipses. Note also that $[-1,1]$ is *not* a level curve for the former while it *is* for the latter.

Figure 3-11 Comparison of equi-potential curves for *Top*: degree 17 monic polynomial interpolant to 17 equispaced zeros and *Bottom*: degree 17 monic polynomial interpolant to 17 zeros distributed according to Chebyshev Gauss nodes resulting in Chebyshev polynomial of degree 17.

3.12 APPROXIMATION ERROR OF POLYNOMIAL SPECTRAL METHODS: TRUNCATION PLUS ALIASING REVISITED

As with trigonometric approximations, polynomial approximations involve two kinds of error. Truncation error is the error incurred when we approximate an infinite continuous expansion by a finite continuous expansion. It can be shown that for any function in $H'_w[-1,1]$ there exists a constant C , independent of N , such that

$$\|u - P_N u\|_{H_w^q[-1,1]} \leq C \|u\|_{H_w^r[-1,1]} N^{\sigma(r,q)},$$

$$\sigma(r,q) = \begin{cases} \frac{3}{2}q - r & 0 \leq q \leq 1, \\ 2q - r - \frac{1}{2} & q \geq 1. \end{cases} \quad (3.73)$$

where $0 \leq q \leq r$. This is a bound on truncation error. In energy norm this bound simplifies to

$$\|u - P_N u\|_{L_w^2[-1,1]} \leq C \|u\|_{H_w^r[-1,1]} N^{-r} \quad r \geq 0 \quad (3.74)$$

which again signals spectral decay of the truncation error.

The second error incurred is when we approximate the continuous expansion coefficients using quadratures. This is the aliasing error: $A_N u(x) = P_N u(x) - I_N u(x)$. Similar to the trigonometric case we have

$$\tilde{u}_n = \hat{u}_n + \frac{1}{\tilde{\gamma}_n} \sum_{m>N}^{\infty} [P_m^{(\alpha)}, P_n^{(\alpha)}]_w \hat{u}_n \quad (3.75)$$

which makes the aliasing error

$$A_N u(x) = \sum_{k=0}^N \left(\frac{1}{\tilde{\gamma}_n} \sum_{m>N}^{\infty} [P_m^{(\alpha)}, P_n^{(\alpha)}]_w \hat{u}_n \right) P_n^{(\alpha)}(x) \quad (3.76)$$

$$= \sum_{m>N}^{\infty} \hat{u}_m I_N P_m^{(\alpha)}$$

It is seen from Eq. (3.76) that aliasing is caused by using the interpolation of the basis rather than the basis itself to represent the higher modes [26]. For discrete Chebyshev expansion the inner products simplify to

$$\tilde{p}_n = \frac{1}{\tilde{\gamma}_n} [P_m^{(\alpha)}, P_n^{(\alpha)}]_w = \begin{cases} 1 & m = 2Np \pm np = 0, \pm 1, \dots, \\ 0 & \text{otherwise.} \end{cases} \quad (3.77)$$

This in turn simplifies the aliasing error to

$$A_N u(x) = \sum_{m>N}^{\infty} \hat{u}_m I_N T_m(x) = \sum_{p=-\infty}^{\infty} \left(\hat{u}_{2Np+n} + \hat{u}_{2Np-n} \right) T_n(x). \quad (3.78)$$

Combining Eq. (3.78) and Eq. (3.73), the approximation error of discrete Chebyshev expansion becomes

$$\|u - I_N u\|_{H_w^q[-1,1]} \leq C \|u\|_{H_w^r[-1,1]} N^{2q-r}, \quad (3.79)$$

for any function in $H_w^r[-1,1]$ with $r > 1/2$.

CHAPTER 4. FINITE DIFFERENCE METHODS IN 1D DOMAINS

4.1 RATIONALE: VALIDATION OF SPECTRAL IDENTIFICATION

On the face of it, presenting finite differences might seem superfluous, as this work pivots around spectral methods. However, finite differences are not only important in their own right they are also used as a “test bed” for validating the spectral results. The ability to generate finite difference stencils for arbitrarily spaced grids is crucial as polynomial spectral methods are based on non-equispaced grids and equispaced finite difference stencils already available will not be applicable on such grids. Furthermore, the need for changing the number of grid points stipulates that we be able to update the finite difference weights accordingly, a task infeasible if left to classical means.

The keystone of finite difference schemes is to replace derivatives with linear combinations of function values. Saying that might cause confusion, as that was the central theme of pseudospectral methods too. To discriminate the two we should note that finite differences employ *overlapping low-order* polynomials to interpolate the function *locally* and then differentiate the interpolant to approximate the derivative at the point at hand. Collocation methods, on the other hand, use a *single high-order* polynomial interpolant for the whole domain whose differentiation then approximates the derivative.

Another interpretation of finite differences that is of great importance is that they are local approximations to differential operators. We opt to commence rigorously based on this interpretation as it has the redeeming virtue of ease of applicability on equispaced grids.

4.2 FINITE DIFFERENCE OPERATORS AND WEIGHTS ON EQUISPACED GRIDS

It is customary to present finite differences on $\mathbf{z} = \{z_k\}_{k=-\infty}^{\infty}$, an *infinite* complex sequence.

The results can be expressed for *finite* sequences straightforwardly, though not neatly.

The following basic operators are defined as acting on every element of $\mathbf{z}$:

shift	$\mathcal{E}z_k = z_{k+1}$	
forward difference	$\Delta_+ z_k = z_{k+1} - z_k$	
backward difference	$\Delta_- z_k = z_k - z_{k-1}$	(4.1)
central difference	$\Delta_0 z_k = z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}$	
average	$\Upsilon_0 z_k = \frac{1}{2}(z_{k+\frac{1}{2}} + z_{k-\frac{1}{2}})$	

If we relate the sequence $\mathbf{z}$ to the samples of a function z and assume, for now, that the samples are spaced an equal distance h apart, and that the function is entire, we can define a differential operator that yields the derivative of z at the k th grid point

$$(D\mathbf{z})_k = z'(kh) \quad (4.2)$$

The goal is now to dispense with D and represent it using the basic operators of Eq. (4.1); this is the gist of finite difference schemes.

4.2.1 TAYLOR APPROXIMATION

The idea is to first define general functions of the basic operators (or linear combinations thereof, as required) since, as we shall see shortly, *before* approximation, the differential operator is *not* a linear function of the *basic* finite difference operators. Defining functions of operators pivots around Taylor expansion which is convergent if we are “close enough” to the expansion point. We note that $\mathcal{E} - I, \Upsilon_0 - I, \Delta_+, \Delta_-, \Delta_0, hD$ all approach the zero operator when the grid spacing, h , approaches zero. As a result we can

formally expand a function in terms of these operators. The result is what we sought - a function of that operator. Then we can operate by that expansion on our sequence. For instance, if $g(x)$ is the function we want to combine Δ_+ with, then following the outline just discussed, we obtain

$$g(\Delta_+)z = \left(\sum_{j=0}^{\infty} a_j \Delta_+^j \right) z = \sum_{j=0}^{\infty} a_j (\Delta_+^j z). \quad (4.3)$$

It is standard practice to express all operators in the currency of the shift operator [42].

This is trivial for forward and backward difference operators; $\Delta_+ = \varepsilon - I$ and $\Delta_- = I - \varepsilon^{-1}$.

Based on Eq. (4.3) we can deduce that $\varepsilon^{1/2}$ is a “half-shift” operator and that implies

$$\Delta_0 = \varepsilon^{1/2} - \varepsilon^{-1/2} \quad \text{and} \quad \Upsilon_0 = \frac{1}{2}(\varepsilon^{1/2} + \varepsilon^{-1/2}).$$

It remains to “change the currency” for the

differentiation operator D .

$$\begin{aligned} \varepsilon z(x) &= z(x+h) = \sum_{j=0}^{\infty} \frac{1}{j!} \frac{d^j z(x)}{dx^j} h^j = \sum_{j=0}^{\infty} \frac{1}{j!} (hD)^j z(x) \\ &= e^{hD} z(x), \end{aligned} \quad (4.4)$$

from which we conclude $\varepsilon = e^{hD}$ or

$$hD = \ln(\varepsilon). \quad (4.5)$$

We are now in a place to express the differential operator D in terms of the difference operators. Expressions in terms of forward and backward difference operators are again trivial but nevertheless important. They are

$$hD = \ln(\Delta_+ + I), \quad (4.6)$$

and

$$hD = -\ln(I - \Delta_-). \quad (4.7)$$

Expressing the differential operator D in terms of the central difference operator Δ_0 requires “solving” $\Delta_0 = \varepsilon^{1/2} - \varepsilon^{-1/2}$ for the shift operator which gives

$$\varepsilon = \left(\frac{1}{2} \Delta_0 + \sqrt{I + \frac{1}{4} \Delta_0^2} \right)^2 \quad (4.8)$$

and that in turn yields

$$hD = 2 \ln \left(\frac{1}{2} \Delta_0 + \sqrt{I + \frac{1}{4} \Delta_0^2} \right). \quad (4.9)$$

The formulas just derived for D are readily expanded and based on the order of approximation required a finite number of terms are retained and the rest of the series is truncated.

For example, obtaining an $O(h^3)$ approximation to the first order differential operator based on forward differences is a two-liner:

$$\begin{aligned} D &= \frac{1}{h} \ln(I + \Delta_+) = \frac{1}{h} \left(\Delta_+ - \frac{1}{2} \Delta_+^2 + \frac{1}{3} \Delta_+^3 + O(\Delta_+^4) \right) \\ &= \frac{1}{h} \left(\Delta_+ - \frac{1}{2} \Delta_+^2 + \frac{1}{3} \Delta_+^3 \right) + O(h^3), \quad h \rightarrow \infty. \end{aligned} \quad (4.10)$$

This makes short work of finding approximations to higher derivatives. For instance, to get the 4th order differential operator approximated up to $O(h^6)$ based on forward differences, we either expand Eq. (4.6) up to Δ_+^{4+6-1} , apply it 4 times, and ignore the terms depending on Δ_+^{4+6} and higher, or expand $\frac{1}{h^4} \ln^4(I + \Delta_+)$ up to order 9. Either way the result is

$$D^4 = \frac{1}{h^4} \left(\Delta_+^4 - 2\Delta_+^5 + \frac{17}{6} \Delta_+^6 - \frac{7}{2} \Delta_+^7 + \frac{967}{240} \Delta_+^8 - \frac{89}{20} \Delta_+^9 \right) + O(h^6), \quad h \rightarrow \infty. \quad (4.11)$$

The natural question that must have been conjured up by now is whether it is reasonable at all to use forward or backward differences. That is, why use values only to the left or right of the point of interest when we can use values from both sides? For one thing, maybe we cannot! We draw attention to the fact that although we have assumed our data sequence to be defined on an infinite grid, realistic grids are *finite*; they have *boundaries*. Generally speaking, near or on the boundary we are bound to use one-sided differences as centered ones would require operating somewhere out of reach.

The approximation of the differential operator based on central difference operators as given in Eq. (4.9) is problematic; expansion leads to a series in which all powers of the central difference operator are odd

$$D = \frac{2}{h} \sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} \binom{2j}{j} \left(\frac{1}{4} \Delta_0 \right)^{2j+1}. \quad (4.12)$$

Even derivatives will obviously and fortuitously bypass this problem as raising an odd power to an *even* power yields an even power. For instance, the $O(h^6)$ approximation to the 4th order differential operator based on central differences becomes

$$D^4 = \frac{1}{h^4} (\Delta_0^4 - \frac{1}{6} \Delta_0^6 + \frac{7}{240} \Delta_0^8) + O(h^6) \quad (4.13)$$

To resolve the problem for odd derivatives, we express the averaging operator in terms of central difference operator, $\Upsilon_0 = (I + \frac{1}{4} \Delta_0^2)^{1/2}$. This gives

$$I = \Upsilon_0 (I + \frac{1}{4} \Delta_0^2)^{-1/2} = \Upsilon_0 \sum_{j=0}^{\infty} (-1)^j \binom{2j}{j} \left(\frac{1}{16} \Delta_0^2 \right)^j, \quad (4.14)$$

the identity operator in disguise. Multiplying Eq. (4.12) by Eq. (4.14) does the trick

$$D = \frac{1}{h} \Upsilon_0 \Delta_0 \left[\sum_{j=0}^{\infty} (-1)^j \binom{2j}{j} \left(\frac{1}{16} \Delta_0^2 \right)^j \right] \left[\sum_{j=0}^{\infty} \frac{(-1)^j}{2j+1} \binom{2j}{j} \left(\frac{1}{4} \Delta_0 \right)^{2j} \right]. \quad (4.15)$$

Since $\Upsilon_0 \Delta_0 z_k = \Upsilon_0 (z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}) = \frac{1}{2} (z_{k+1} - z_{k-1})$ and the rest of the equation, although daunting, contains only even powers of the central difference operator, it follows that Eq. (4.15) comprises terms that live on the grid, as opposed to Eq. (4.12) that would try in vain to operate in the nowhere land; it is comprised of terms that reside half-way between the grid points – points that are nonexistent. Raising Eq. (4.15) to a power is possible but not recommended as it would require more grid points than necessary. To attain approximants to differential operators of odd order, Eq. (4.12) is raised to the odd power required and then multiplied by the contrived identity given by Eq. (4.14) to bring the terms on the grid. As an example, employing this procedure, the $O(h^6)$ approximation to the 5th order differential operator based on central differences becomes

$$D^5 = \frac{1}{h^5} \Upsilon_0 \Delta_0 (\Delta_0^4 - \frac{1}{3} \Delta_0^6 + \frac{13}{144} \Delta_0^8) + O(h^6). \quad (4.16)$$

We cannot overemphasize this point that in none of the example approximations given have we used the (formidable-seeming) series representations that were employed for proving the procedure. Expansions of the relevant functions of operators were implemented *symbolically* in a single line command producing the corresponding approximants to the differential operators. That is the power of the method presented.

4.2.2 PADE APPROXIMATION AND A SYMBOLIC METHOD FOR CALCULATING WEIGHTS

So far we have been dealing with differential operators approximated in terms of difference operators. This has the advantage of producing compact expressions for

approximating high order differential operators to high orders of accuracy. Simple routines can then be coded to implement the basic finite difference operators and attain the approximation to the differential operators in the end. On the other hand, sometimes it is more convenient to have expressions in term of the grid points z_k rather than ones involving operators acting on the grid points. Theoretically conversion from one representation to another is trivial but practically we need to have both representations at hand especially for our case that deals with different orders of differentiation and accuracy.

From a different prospective, we need to be able to deal with implicit cases too. Thus a slightly streamlined procedure than the one presented in the previous section is in order.

We present an example to illustrate a more general method based on Pade approximations [43]. Consider an example of Collatz “Mehrstellenverfahren”. The weights b_i, c_i are to be found such that the numerical molecule

$$b_0 u''(x-h) + b_1 u''(x) + b_2 u''(x+h) \approx c_0 u(x-h) + c_1 u(x) + c_2 u(x+h) \quad (4.17)$$

has the highest accuracy possible. If we take the Fourier transform of Eq. (4.17) and note that $\hat{z}(x+h) = \widehat{\varepsilon z(x)} = e^{ih\xi} \hat{z}(\xi)$, we obtain

$$-\xi^2 (b_0 e^{-ih\xi} + b_1 + b_2 e^{ih\xi}) \approx c_0 e^{-ih\xi} + c_1 + c_2 e^{ih\xi}. \quad (4.18)$$

Consequently,

$$\left(\frac{\ln(e^{ih\xi})}{h} \right)^2 \approx \frac{c_0 + c_1 e^{ih\xi} + c_2 (e^{ih\xi})^2}{b_0 + b_1 e^{ih\xi} + b_2 (e^{ih\xi})^2} \quad (4.19)$$

Pade approximation [44] of the left hand side around 1 to order [2,2] offers the highest accuracy that can be provided by this rational function:

$$\left(\frac{\ln(e^{ih\xi})}{h} \right)^2 \approx \frac{12 - 24e^{ih\xi} + 12(e^{ih\xi})^2}{h^2(1 + 10e^{ih\xi} + 1(e^{ih\xi})^2)}. \quad (4.20)$$

Comparison of Eq. (4.20) and Eq. (4.19) determines the coefficients. Figure 4-1 plots both sides of Eq. (4.20). Note the superb accuracy of the Pade approximant around 1 which can be explained by comparing the Taylor expansion of $\ln(x)^2$ around $x = 1$,

$$\begin{aligned} \text{Taylor } \ln(x)^2|_{x=1} &= (x-1)^2 - (x-1)^3 + \frac{11}{12}(x-1)^4 - \frac{5}{6}(x-1)^5 \\ &\quad + \frac{137}{180}(x-1)^6 + O((x-1)^7), \end{aligned} \quad (4.21)$$

and that of the Pade approximant around $x = 1$,

$$\begin{aligned} \text{Taylor } \frac{12 - 24x - 12x^2}{1 + 10x + x^2}|_{x=1} &= (x-1)^2 - (x-1)^3 + \frac{11}{12}(x-1)^4 - \frac{5}{6}(x-1)^5 \\ &\quad + \frac{109}{144}(x-1)^6 + O((x-1)^7). \end{aligned} \quad (4.22)$$

Notice the agreement of the two up to order $2+2+1=5$. This is not serendipitous and is exactly a consequence of the way a Pade approximant is defined; that agreement was guaranteed.

This procedure carries over to staggered (but still regular) grids too. If there are s grid intervals between the leftmost derivative and function entries, with positive s if the former is to the right of the latter and negative s the other way round, d is the number of grid intervals between the left- and right-most derivative entries, and n is the same count for the function entries, then a $[n, d]$ Pade approximant around $x = 1$ to $x^s \ln(x)^m$ yields the finite difference coefficients for approximating m th derivative in a general (i.e., implicit) expression [43].

Figure 4-1 $\ln(x)^2$ and its [2,2] Pade approximant around 1. $\log$ and $\ln$ both refer to natural logarithm.

Figure 4-2 The implicit regular staggered stencil of Eq. (4.23); leftmost derivative entry to the right (+) of the leftmost function entry

We close this section on a high note: calculating the b_i, c_i 's that make the following *implicit regular staggered* stencil optimal

$$b_0 u^{(4)}(x-h) + b_1 u^{(4)}(x) + b_2 u^{(4)}(x+h) + b_3 u^{(4)}(x+2h) \approx c_0 u(x-\frac{3}{2}h) + c_1 u(x-\frac{1}{2}h) + c_2 u(x+\frac{1}{2}h) + c_3 u(x+\frac{3}{2}h) + c_4 u(x+\frac{5}{2}h). \quad (4.23)$$

Figure 4-2 illustrates this stencil schematically. Applying the Pade weight algorithm outlined above we obtain 1, 23, 23, 1 for b_0 through b_3 and $(48, -192, 288, -192, 48)/h^4$ for c_0 through c_4 .

Figure 4-3 The implicit regular staggered stencil of Eq. (4.24); leftmost derivative entry to the left (-) of the leftmost function entry

Note that if the stencil was as depicted in Figure 4-3 it would be related to

$$b_0 u^{(4)}(x-h) + b_1 u^{(4)}(x) + b_2 u^{(4)}(x+h) + b_3 u^{(4)}(x+2h) \approx c_0 u(x+\frac{1}{2}h) + c_1 u(x+\frac{3}{2}h) + c_2 u(x+\frac{5}{2}h) + c_3 u(x+\frac{7}{2}h) + c_4 u(x+\frac{9}{2}h), \quad (4.24)$$

and we would obtain $-27, 107, -157, 125$ for b_0 through b_3 and again $(48, -192, 288, -192, 48)/h^4$ for c_0 through c_4 .

There is an accuracy gain in staggering. Suppose we are to approximate the first order derivative using the stencil of Figure 4-4, which is explicit, regular, and *non-staggered*,

Figure 4-4 An explicit, *non-staggered* stencil for approximation of first derivative

the coefficients will be $(2, -16, 0, 16, -2)/24$. On the other hand, on the explicit, regular, and *staggered* stencil of Figure 4-5 the coefficients become $(1, -27, 27, -1)/24$. It is clear that the “staggered coefficients” are more local and that makes sense in the light of the fact that the derivative, being the slope of the tangent at the point of interest, is also a local feature of a function.

Figure 4-5 An explicit, regular, and *staggered* stencil for approximation of first derivative

4.3 FINITE DIFFERENCE WEIGHTS ON ARBITRARILY SPACED GRIDS

Up until now all our discussions of finite differences were restricted to equispaced grids. We could have explicit or implicit expressions and the grid could be staggered but the spacing was always fixed from point to point. As general as the expression “implicit regular staggered stencil” might sound, it is still one step behind the most general a stencil can get. The question is whether *we* need anything more general than that and

without answering that it might seem as if this section had arisen from mathematical nitpicking or infatuation with abstraction. The answer is an emphatic yes. For one thing, we embarked on this chapter with the aim of introducing finite difference methods as a validation tool for checking the results produced by spectral methods and the polynomial spectral methods live on *irregular* grids.

We have discussed the polynomial cardinal function $\ell_j(x)$ and the resulting Lagrange form of interpolating polynomial $p_N(x)$; the polynomial was of degree N , the highest possible as we used *all* the grid points $x_j, j=0,1,\dots,N$. That was for polynomial pseudospectral methods; we sought a single *global* polynomial interpolant extending on the whole grid. For finite differences we can but usually do not go for the highest degree polynomial. Instead we make use of lower order ones. Thusly, we need one little tweak. Interpolation will be not only on the full set $\{x_j, j=0,1,\dots,N\}$ but also on all leading subsets $\{x_j, j=0,1,\dots,k\}, k=0,1,\dots,N$ and the Lagrange form of the interpolating polynomial would be

$$\begin{aligned} p_k(x) &= \sum_{j=0}^k \ell_{j,k}(x) u_j, \\ \ell_{j,k}(x) &= \frac{\prod_{i=0, i \neq j}^k (x - x_i)}{\prod_{i=0, i \neq j}^k (x_j - x_i)}. \end{aligned} \tag{4.25}$$

Without loss of generality it is assumed that the approximations to derivatives are to be optimal at $x=0$. Differentiation of Eq. (4.25) yields

$$\begin{aligned} u^{(m)}(x)|_{x=0} &= \frac{d^m p_k(x)}{dx^m} \Big|_{x=0} = \sum_{j=0}^k \frac{d^m \ell_{j,k}(x)}{dx^m} \Big|_{x=0} u_j \\ &= \sum_{j=0}^k c_{j,k}^{(m)} u_j. \end{aligned} \tag{4.26}$$

We can expand the cardinal function around zero using Taylor's theorem to get

$$\ell_{j,k}(x) = \sum_{m=0}^k \frac{d^m \ell_{j,k}(x)}{dx^m} \Big|_{x=0} \frac{x^k}{k!} = \sum_{m=0}^k c_{j,k}^{(m)} \frac{x^k}{k!}. \quad (4.27)$$

Eq. (4.27) gives the cardinal function as a polynomial based on powers of x . Eq. (4.25) defines the cardinal function based on factors of $(x - x_i)$. As these two equations are two different representations of one polynomial, finding the coefficients $c_{j,k}^{(m)}$ in Eq. (4.27) is equivalent to a re-arranging of the polynomial cardinal $\ell_{j,k}(x)$ in Eq. (4.25). The proper rearranging can be neatly attained through the use of the following two recursion relations discovered by Fornberg [45]

$$\ell_{j,k}(x) = \frac{(x - x_k)}{(x_j - x_k)} \ell_{j,k-1}(x), \quad (4.28)$$

and

$$\ell_{k,k}(x) = \frac{\prod_{v=0}^{k-2} (x_{k-1} - x_v)}{\prod_{v=0}^{k-1} (x_k - x_v)} (x - x_{k-1}) \ell_{k-1,k-1}(x) \quad (4.29)$$

which, after substituting Eq. (4.27) for the polynomial cardinal, in turn yield the recursion relations for the finite difference weights

$$\begin{aligned} c_{j,k}^{(m)} &= \frac{1}{x_k - x_j} \left(x_k c_{j,k-1}^{(m)} - (m) c_{j,k-1}^{(m-1)} \right) \\ c_{k,k}^{(m)} &= \frac{\prod_{v=0}^{k-2} (x_{k-1} - x_v)}{\prod_{v=0}^{k-1} (x_k - x_v)} \left((m) c_{k-1,k-1}^{(m-1)} - x_{k-1} c_{k-1,k-1}^{(m)} \right) \end{aligned} \quad (4.30)$$

CHAPTER 5. IDENTIFICATION OF SPATIOTEMPORAL DIFFERENTIAL EQUATIONS

Spatiotemporal differential equations are well capable of describing a wide variety of phenomena and, as mentioned before, deriving them based on first principles is not usually feasible and data-driven modeling (otherwise known as system identification) comes into picture indispensably.

In all of the identification problems presented here, simulated data is used and that categorizes these case studies as *numerical experiments*; as such, we are certain that all components of the outputs are *observable* at the location(s) required by our identification procedures. In practice, however, this might not be the case and some state-space *reconstruction* may be required [46], [47], [48].

Direct solutions of the example problems presented here are based on the so called “method of lines” [42] starting with spectral discretization of the space dimension. The time marching is performed with exponential time differencing fourth-order Runge-Kutta (ETDRK4) [49].

Our system identification experiments may be categorized based on the class of spectral operators they require for approximating the derivatives in the model structure Eq. (1.11), namely trigonometric or algebraic. We have developed both classes of operators and hence are equipped with the tools required for approximating the model structure using a finite set of sample data. This will allow us to arrive at a discretization of that continuous

structure, which is then fed to the estimation and system simplification algorithms to reconstruct the true model.

We presented in CHAPTER 2, CHAPTER 3, and CHAPTER 4 the material required for constructing several derivative approximation operators. For the sake of generality, let us assume in the present discussion that the *appropriate* operators for the approximating the derivatives in Eq. (1.11) are chosen. For the system identification problems to be presented, we will then choose specific derivative operators.

Let us start by approximating the time derivative. Let V denote a matrix that contains in each *column* j a grid function $v|_{t_j}$ sampling the function $u(x, t)$ over all the quadrature nodes x_i at time t_j , and let $D_t^{(m)}$ denote the matrix corresponding to an appropriate linear operator for approximating the time derivative $\partial^{(m)}u(x, t)/\partial t^{(m)}$ at the node x_i . As each *row* of V samples the time evolution of u at the node x_i , $V^{(m)} = (D_t^{(m)}V^T)^T = VD_t^{(m)}$ will be a matrix approximating $\partial^{(m)}u(x, t)/\partial t^{(m)}$. As with V , each column j of $V^{(m)}$ approximates $\partial^{(m)}u(x, t)/\partial t^{(m)}$ over all the quadrature nodes x_i at time t_j . Let $\hat{Y}$ denote a vector of size $N' \times 1$ ($C < N' \leq N$) containing N' elements of a column of $V^{(m)}$ corresponding to an appropriate time point t_{sysid} for identifying the system. Clearly, $\hat{Y}$ contains an approximation to $\partial^{(m)}u(x, t)/\partial t^{(m)}$ at time t_{sysid} over N' quadrature nodes. There is no need for these N' elements to be chosen contiguously.

It remains to approximate the spatial derivatives. Applying an appropriate (q) -th order spatial differentiation matrix $D^{(q)}$ to the grid function $v|_{t_{\text{sysid}}}$ (which samples $u(x, t)$ at

time t_{sysid} over all the quadrature nodes x_i) yields an approximation $v^{(q)}|_{t_{\text{sysid}}}$ of $\partial^q u(x, t) / \partial x^q$ at that fixed time point: $v^{(q)} = D^{(q)} v|_{t_{\text{sysid}}}$.

Let Φ denote the design matrix containing in each column j an approximation, at time t_{sysid} and over the same N' quadrature nodes as for $\hat{Y}$, to the kernel $x^{p_1} u^{p_2} (\partial u / \partial x)^{p_3} (\partial^2 u / \partial x^2)^{p_4} \dots (\partial^n u / \partial x^n)^{p_{n+2}}$ corresponding to the parameter θ_j in Eq. (1.11). Then the *continuous* model structure of Eq. (1.11) can be approximated by an equivalent *discrete* regression form given in Eq. (1.12). In CHAPTER 1 we discussed how to estimate the parameters of that regression form.

We now present several identification problems and their solutions to demonstrate the spectral identification methodology introduced in this work.

5.1 IDENTIFICATION OF KURAMOTO-SIVASHINSKY EQUATION

Kuramoto-Sivashinsky (KS) equation was derived by Kuramoto and, independently, Sivashinsky; the former derived it in the context of angular-phase turbulence for a particular reaction-diffusion system [50, 51] and the latter in modeling small thermal diffusive instabilities in laminar flame fronts [52, 53]. In general, it models pattern formations and is an instance of low-dimensional behavior in PDEs [54]. It is characterized by the presence of coherent spatial structures and temporal chaos [55].

We consider the identification of the following one dimensional KS equation

$$\begin{aligned} u_t &= -uu_x - u_{xx} - \varepsilon u_{xxxx}, \quad (x, t) \in I \times \mathbb{R}_+, \\ u(x, 0) &= u_0(x), \quad u(x + L, t) = u(x, t), \end{aligned} \tag{5.1}$$

where the subscripts denote partial differentiation, ε is a positive parameter, and u_0 is of period L (the size of a typical pattern cell). In this particular example we pick $\varepsilon = 1$,

$I = [0, L)$, $L = 32\pi$, and $u_0(x) = \cos(x/16)(1 + \sin(x/16))$. Figure 5-1 shows the solution to Eq. (5.1) for this ε and these initial and boundary conditions. This is an example of deterministic chaos [56].

Figure 5-1 Solution of KS Eq. (5.1) up to $t = 150$

We compare the performance of system identification using finite differences and Fourier spectral methods. The time derivative was approximated using centered finite differences of 4th order. Of course, for finite difference identification that same operator was employed to approximate the *spatial* derivatives too. For the spectral identification, spatial derivatives were approximated using Fourier spectral operators. To obtain the regression matrix, the state of the system was then picked at $N = 200$ space points at a fixed time point. We have assumed a degree of nonlinearity of $DoNL = 2$ and a highest

order of spatial derivative of $n=4$ for the model. Therefore, there are a total of 28 monomials in the model among which 25 monomials are redundant. All of the monomials, redundant or not, are listed in the first column of Table 5-1. To generate the data for identification, the direct problem was solved, with $N=200$ grid points in space up to $t=150$, with time steps of $\Delta t=0.1$ and $\Delta t=0.01$. Table 5-1 compares the parameter estimates with and without system simplification using the time steps $\Delta t=0.1$ and $\Delta t=0.01$ respectively, using finite differences and Fourier spectral identification. Scrutinizing the results, we notice three interesting points. (i) For both finite differences and Fourier spectral identification, the technique of system simplification successfully eliminates most redundant monomials, making the general identification model approach the true description the original system. More specifically, for the case of spectral identification, when applying a 0.5% ERR threshold, all redundant terms are removed except for the linear term u . (ii) Whether using the system simplification technique or not, all the test cases show that the spectral identification outperforms, significantly, the finite differences in terms of accuracy. (iii) When the time step Δt is made smaller, the parameter estimates generally become more accurate for the spectral identification. However, for the finite differences case, a peculiar phenomenon was observed: the parameter estimates have no improvement with no system simplification (*No SS*), and become even worse after applying the system simplification (*SS*). With *No SS*, using finite differences, we believe parameter estimate errors are induced mainly by spatial derivative approximations (rather than time derivative approximations). For the case *with SS*, the deterioration, which we call “backfiring” phenomenon, is because the system

simplification imposes additional constraints (namely setting some of the parameters to zero) on the optimization of parameter estimates.

Case	$\Delta t = 0.1$; No SS		$\Delta t = 0.1$; with SS		$\Delta t = 0.01$; No SS		$\Delta t = 0.01$; with SS	
Terms	Finite Diff	Spectral	Finite Diff	Spectral	Finite Diff	Spectral	Finite Diff	Spectral
1	0.00014898	0.00000189	0	0	0.00013500	0.00000036	0	0
x	-0.00001028	0.00000002	0	0	-0.00000988	-0.00000002	0	0
u	-0.00016457	-0.00000753	0.0017379	1.922E-07	-0.00013566	-0.00000064	0.001964623	2.49966E-08
u_x	-0.00017899	0.00000203	0	0	-0.00020427	0.00000028	0	0
u_{xx}	-1.00710797	-1.00003295	-0.951403	-1	-1.00699037	-1.00000309	0.952312176	1.000000014
u_{xxx}	0.00024672	0.00000438	0	0	0.00023069	0.00000025	0	0
u_{xxxx}	-1.00793344	-1.00001526	-0.97414	-1.000001	-1.00798305	-1.00000129	0.974721479	1.000000014
x^2	0.00000010	0.00000000	0	0	0.00000010	0.00000000	0	0
xu	-0.00000610	0.00000022	0.0001011	4.169E-09	-0.00000657	0.00000001	9.58702E-05	-4.75549E-10
xu_x	0.00000304	0.00000006	0	0	0.00000306	0.00000000	0	0
xu_{xx}	-0.00000298	0.00000081	0	0	-0.00000652	0.00000002	0	0
xu_{xxx}	0.00000155	0.00000004	0	0	0.00000150	0.00000000	0	0
xu_{xxxx}	0.00002245	0.00000047	0	0	0.00002153	0.00000000	0	0
u^2	-0.00004683	-0.00000176	0	0	-0.00003492	-0.00000011	0	0
uu_x	-1.00445988	-0.99998402	-1.048316	-1.000003	-1.00442362	-0.99999803	1.048476897	1.000000016
uu_{xx}	-0.00054926	-0.00001235	0	0	-0.00045845	-0.00000078	0	0
uu_{xxx}	-0.00369895	0.00002857	-0.063258	0	-0.00358734	0.00000222	-0.06298254	0
uu_{xxxx}	-0.00063807	-0.00000477	0	0	-0.00062970	-0.00000017	0	0
u_x^2	-0.00032889	-0.00000820	0	0	-0.00033534	0.00000025	0	0
$u_x u_{xx}$	-0.01187819	-0.00001102	-0.141359	0	-0.01154527	0.00001040	0.141473438	0
$u_x u_{xxx}$	-0.00032100	-0.00001925	0	0	-0.00030879	0.00000009	0	0
$u_x u_{xxxx}$	0.04456847	-0.00000394	0	0	0.04472038	0.00000547	0	0
u_{xx}^2	-0.00007835	-0.00000253	0	0	0.00003172	-0.00000076	0	0
$u_{xx} u_{xxx}$	0.09621567	0.00004008	0	0	0.09646444	0.00000637	0	0
$u_{xx} u_{xxxx}$	-0.00075360	-0.00000695	0	0	-0.00070704	-0.00000038	0	0
u_{xxx}^2	-0.00017271	-0.00001076	0	0	-0.00015949	-0.00000008	0	0
$u_{xxx} u_{xxxx}$	0.02495664	0.00001295	0	0	0.02506517	0.00000389	0	0
u_{xxxx}^2	-0.00019752	-0.00000295	0	0	-0.00018786	-0.00000008	0	0

Table 5-1 Comparison of parameter estimates in identification of KS equation, using finite differences and Fourier spectral methods, with and without system simplification (SS). ERR threshold 0.5%; spatial samples $N=200$; and the state data picked at the 400th time point for the case of $\Delta t = 0.1$, and at the 4000th time point for the case of $\Delta t = 0.01$.

These constraints leave fewer variables available for compensating the errors induced by spatial derivative approximations. This, in turn, results in some accuracy sacrifice in the parameter estimates causing more errors in them than the unconstrained parameter

estimates (i.e. with *No SS*), as mentioned before in the discussion of ERR technique. Here, the key fact is that the finite differences cannot generate better spatial derivative approximations than spectral methods. The “backfiring” phenomenon of the finite differences persists in many of our test cases.

Another advantage of spectral identification is the less number of samples they require for accurate identification. This is illustrated in Figure 5-2a by the 2-norm of parameter estimate error for increasing number of samples, whose states were picked at the 400th time point. Less error for the same number of samples and much faster decay of error are clearly seen for the spectral identification. Also note that “paying” more samples to finite differences cannot “buy” the accuracy of spectral identification unless we are willing to pay a fortune! Particularly, in Figure 5-2a, the accuracy we get with spectral identification with 128 samples could not be caught up by finite differences even if we provided 200 samples.

A point regarding time points is in order now. The choice of the time point at which to pick the state data can have a certain effect on the identification results; the system is evolving with time and at a particular time point, state variables maybe be so small that picking them can cause a rank deficiency in the regression matrix used in least squares problem. To avoid this issue, in an actual identification experiment the input signal should be chosen to be *persistently exciting* [57]. Other than that rank deficiency problem, which we do not address further here, we would like to illustrate the certain effect of the time point choice on both finite difference and spectral identifications, in Figure 5-2a (state data picked at 400th time point) and Figure 5-2b (state data picked at 200th time point). To find out if the time point choice could seriously affect the

reconstruction results in the sense of the accuracy of the identified parameters, we directly investigated the variation of the accuracy of the identified parameters of the KS system with different time point choices via a systematic calculation for both methods. Figure 5-3 illustrates the effect of the time point choice on relative error in the coefficients of the dominant terms, namely u_{xx} , u_{xxxx} , and uu_x , of the reconstructed KS system. Figure 5-3b indicates that our proposed spectral identification is *not* “sensitive” to the time point choice as the relative error in the identified parameters does not exceed 0.05%. On the other hand, finite difference identification is relatively sensitive to the time point choice, as the corresponding relative errors fluctuate as much as 26%, as shown in Figure 5-3a. For our spectral identification the time point choice in the interval [250,400] results in better accuracy than other time point choices, but the difference in accuracy is insignificant.

In Figure 5-4 we have highlighted the positive effect of the system simplification on the decrease of the parameter estimate error as well as, again, the superiority of spectral identification. First of all, it should be noted that spectral identification surpass finite differences even if we improve the latter with system simplification and do not apply that improvement to the former. Secondly, when incorporated in spectral identification, the system simplification decreases the parameter estimate errors by several orders of magnitude. Lastly, the “backfiring” phenomenon is again noticed when the system simplification is applied to finite differences identification.

Figure 5-2 Effect of time point choice on parameter estimate error in identification of KS equation using finite differences and Fourier spectral methods. State data picked at the (a) 400th time point (b) 200th time point. Much less error is incurred with the same number of samples if the time point is chosen “prudently”.

Figure 5-3 Effect of time point choice on relative error in dominant coefficients of KS equation for (a) finite difference and (b) spectral identifications (number of samples $N = 200$, ERR threshold 0.5%). It is seen that the accuracy of the dominant coefficients identified is *not* sensitive to the time point choice for the proposed spectral identification, since the relative error in the identified parameters remains under 0.05%, as shown in (b). However, finite difference identification is relatively sensitive to the time point choice, since the relative error fluctuates as much as 26%, as shown in (a).

Figure 5-4 Comparison of parameter estimate error in identification of KS equation using finite differences and Fourier spectral methods with and without system simplification (ERR threshold 0.5%, state data picked at the 400th time point.) (a) ‘Backfiring’ of system simplification when applied to finite differences resulting in parameter deterioration. (b) Logarithmic scale for error axis emphasizes smaller error (by several orders of magnitude) and much faster decay rate when using spectral methods.

Parameter estimates coming from finite differences identification with system simplification are actually worse than those with no system simplification applied; the terms surviving the system simplification procedure are not accurate enough to give the least sum for squares of prediction errors with the parameter estimates close to the true ones. Therefore, inevitably, parameter estimates deteriorate to give the least squares solution. For spectral identification, however, the remaining terms are accurate enough to give parameter estimates close to the true values; no deterioration occurs.

As a final point, the effect of the selection of error reduction ratio (ERR) threshold on the relative estimate error is examined. By referring to Figure 5-5, we would like to conclude that the final results are insensitive to the threshold selection when the threshold is set to a value in an appropriate interval. Specifically, Figure 5-5 illustrates that the errors remain small and insensitive to the ERR threshold when it is set to a value smaller than 0.01 for the finite difference identification (Figure 5-5a) and 0.008 for the spectral identification (Figure 5-5b). From the *step-like* dependence on the ERR threshold, it is seen that the estimate error attains its minimum in the interval $[0.005, \sim 0.01)$ for finite differences and $[0.0035, \sim 0.008)$ for spectral methods. An ERR threshold in excess of the maximum thresholds mentioned leads to unsatisfactory performance of the structure selection procedure. We also note that reconstructing different systems requires different maximum thresholds. In numerical experiments, we usually seek a proper threshold by starting from zero and increasing the threshold value in small steps until the estimate error reaches a minimum level. Sometimes, trial and error as well as some knowledge of the problem is also helpful in finding a proper threshold. In our numerical experiments, finding a proper threshold for the structure selection has not caused any technical difficulty.

Figure 5-5 Variation of estimate error with different ERR threshold selections for (a) finite difference and (b) spectral identifications (number of samples $N = 200$, state data picked at 400th time point). The estimate error attains its minimum with ERR threshold in $[0.005, \sim 0.01)$ for (a) and $[0.0035, \sim 0.008)$ for (b). Reconstruction process may fail if the threshold is set in excess of a maximum threshold, 0.01 for (a) and 0.008 for (b). A logarithmic scale is used for the lower pane.

5.2 IDENTIFICATION OF BURGERS EQUATION

Burgers equation may be interpreted as a simplified model for turbulence, boundary layer behavior, street traffic, and mass transport, among other phenomena. Its inviscid variant, the nonlinear conservation law, is a prototype for equations which can develop *discontinuities* (shock waves) from *continuous* initial conditions. It is also informative to note that Burgers equation is the 1D Navier-Stokes equation *without* the pressure term and the volume forces. Spectral discretization of Burgers equation raises many issues shared with much more complicated problems and that is the reason we are presenting it as an example.

Burgers equation maybe written as

$$\begin{aligned} u_t &= -uu_x + \nu u_{xx}, \quad (x, t) \in I \times \mathbb{R}_+, \\ u(x, 0) &= u_0(x), \quad u(x + L, t) = u(x, t), \end{aligned} \quad (5.2)$$

where ν is the kinematic viscosity, I is the spatial domain of the problem and $\mathbb{R}_+$ refers to $t > 0$. The boundary conditions are asked to be periodic. For identification, we choose

the initial condition to be $u_0(x) = \exp(-10\sin^2(x/2))$ and the interval to be $I = [-\pi, \pi)$; that is, $L = 2\pi$.

Small viscosities ν cannot erase disturbances and shockwaves would appear. Larger viscosities do not let shocks happen. We have chosen the challengingly small $\nu = 0.03$ to better showcase the superiority of the spectral identification over finite differences.

Table 5-2 gives a comparison of the identified parameters for this equation using finite differences and Fourier spectral identifications. We have chosen to present the parameter estimates *without* any system simplification, to focus on the comparison of finite differences and spectral identifications.

We notice that finite differences identification performs very poorly in this case. This is caused by the low viscosity we have picked and the high wavenumber nature of the resulting shock. Finite differences suffer from *phase errors* and the crests of waves approximated with them are in the wrong places. Spectral identification, on the other hand, has no difficulty in determining the right parameters for the significant terms.

Terms	Parameters	
	Finite Diff	Spectral
1	0.00010	0.00000
x	0.00000	0.00000
u	-0.05340	0.00000
u_x	0.01140	0.00010
u_{xx}	-0.02500	-0.03000
x^2	0.00000	0.00000
xu	0.10800	0.00070
xu_x	0.01060	0.00000
xu_{xx}	-0.00280	0.00000
u^2	0.11210	0.00050
uu_x	-0.98480	-0.99990
uu_{xx}	0.00880	0.00010
u_x^2	-0.00250	-0.00010
$u_x u_{xx}$	0.00230	0.00000
u_{xx}^2	0.00020	0.00000

Table 5-2 Comparison of parameter estimates in reconstruction of Burgers equation, using finite differences and Fourier spectral methods *without* system simplification (SS). Number of spatial samples $N = 68$ and the state data picked at the 5th time point with time step $\Delta t = 0.01$.

The parameter estimate error decays with the increasing the number of samples essentially with the same trend as presented in Figure 5-4.

5.3 IDENTIFICATION OF ALLEN-CAHN EQUATION

Allen-Cahn reaction-diffusion equation demonstrates one of the basic types of nonlinearity known as bistable nonlinearity [58], which, in its simplest form, is caused by a function of the form $u(1-u)(u-\mu)$, where μ is a constant [59].

As our test case, we pick the following Allen-Cahn equation,

$$\begin{aligned} u_t &= \varepsilon u_{xx} + u - u^3, \\ u(-1, t) &= -1, u(1, t) = 1 \end{aligned} \quad (5.3)$$

with ε being a parameter, set to 0.01 in the direct solution.

As depicted in Figure 5-6, the solution of Eq. (5.3) shows three steady states, or three equilibria: $u = -1$, $u = 0$, and $u = 1$. The middle equilibrium is unstable but the side ones are stable and attracting; the solution demonstrates flat surfaces close to the sides with

interfaces in between that may stay intact for a long time before disappearing abruptly. This is metastability, the ability of an unstable equilibrium to persist for a long time scale.

Figure 5-6 Solution of Allen-Cahn Eq. (5.3) up to $t = 70$.

Simulated data from the direct solution (on a spatial grid of $N = 100$ points until $t = 70$ with time steps of $\Delta t = 1/8$) was used for the identification. The time derivative, for both finite differences and Chebyshev spectral identifications, was approximated by centered finite differences of 4th order. Being on a clustered spatial grid, for approximating derivatives using finite differences, we used Fornberg's algorithm for generating finite difference weights on this quadratically spaced grids. For the Chebyshev spectral identification, spatial derivatives were, of course, approximated using the Chebyshev operators. The state of the system was then picked at all of the available grid points at $t = 35$. A degree of nonlinearity of $DoNL = 3$ and a highest order of spatial derivative of $n = 2$ were assumed for the model, producing 35 monomials 32 of which are redundant.

To avoid clutter, we refrain from allowing for the non-existent terms involving x ; that leaves us with 20 terms.

Table 5-3 compares the finite difference and Chebyshev spectral identifications both with and without employing system simplification.

Case	with SS		without SS	
Terms	Finite Diff	Chebyshev	Finite Diff	Chebyshev
1	0	0	0.02773881	0.00000000
u	0.84590495	1.00000181	0.92329432	0.99981599
u_x	0	0	-0.00267348	0.00000000
u_{xx}	0.00817380	0.01000001	0.00915251	0.00999803
u^2	0	0	-0.02770145	0.00000000
uu_x	0.01781328	-0.00000002	0.00052772	-0.00001575
uu_{xx}	0	0	0.00146772	0.00000000
u_x^2	0	0	-0.00106296	0.00000000
$u_x u_{xx}$	0.00016897	0.00000000	0.00002309	-0.00000014
u_{xx}^2	0	0	0.00000359	0.00000000
u^3	-0.84620482	-1.00000188	-0.92319928	-0.99981588
$u^2 u_x$	0.00344708	0	0.01308569	0.00000000
$u^2 u_{xx}$	0.00193432	0	0.00088716	-0.00000110
uu_x^2	0.00308825	0	0.00401588	0.00000859
$uu_x u_{xx}$	0	0	-0.00036433	0.00000000
uu_{xx}^2	0	0	0.00000813	-0.00000003
u_x^3	0	0	0.00000485	0.00000000
$u_x^2 u_{xx}$	0	0	0.00000022	0.00000010
$u_x u_{xx}^2$	0	0	-0.00000080	0.00000000
u_{xx}^3	0	0	-0.00000003	0.00000000

Table 5-3 Comparison of parameter estimates in identification of Allen-Cahn equation, using finite differences and Chebyshev spectral methods, with and without system simplification (SS). ERR threshold 0.1%; number of spatial samples $N = 100$; state data picked at $t = 35$.

Attention is firstly drawn to the excellent performance of the proposed system simplification (SS) algorithm for both finite difference and Chebyshev spectral identifications; parsimony is achieved quite satisfactorily as seen from the *with SS* columns. In fact, with the Chebyshev spectral identification only two redundant terms, namely uu_x and $u_x u_{xx}$ escape our iterative orthogonal filtering if ERR threshold is set to 0.1%. It is also of interest to notice that whether employing SS or not, the Chebyshev spectral identification outperforms, significantly, finite difference identifications in terms

of accuracy. However, for finite difference and Chebyshev spectral identification *with SS* some deterioration is noticed in the “surviving” parameters compared with their corresponding *without SS* parameters. Specifically, the dominant parameters (the coefficients of u, u_{xx} , and u^3) have moved farther from their true values (1, 0.01, and -1, respectively). This is due to the additional constraints imposed by the SS algorithm for the optimization of the parameters. These constraints essentially eliminate some columns of Φ , by eliminating their corresponding $\hat{\theta}_j$, and, after the SS algorithm finishes execution, we are handed over the sub-matrix Φ_{ss} (consisting of the surviving of columns of Φ) and a new set of parameters $\hat{\theta}_{ss}$ given the column *with SS Finite Diff* of Table 5-3. The columns of Φ are calculated based on the spatial derivatives approximated by finite difference methods and that inflicts rather large errors on them. The situation is worse for Φ_{ss} : not only does it inherit the same inaccuracy in its columns, but also it has fewer columns. The least squares criterion has yielded an $\hat{\theta}$ that makes the prediction error $\|Y - \Phi\theta\|_{\ell^2}$ smallest. This $\hat{\theta}$, given in the column *without SS Finite Diff* of Table 5-3, is employing every column of Φ to do its job. On the other hand, $\hat{\theta}_{ss}$ has access to but Φ_{ss} and not the rest of Φ . If $\hat{\theta}_{ss}$ were to equal $\hat{\theta}$ (for the corresponding parameters), the resulting prediction error would not be a minimum. $\hat{\theta}_{ss}$ has to change from $\hat{\theta}$. The finite difference based dominant columns of Φ_{ss} (or Φ) are not accurate enough for $\hat{\theta}_{ss}$ to yield the minimum prediction error employing just them. Therefore, inevitably, $\hat{\theta}_{ss}$ has to heavily use the redundant columns to do the job dictated by the least squares criterion. This phenomenon, which we call “backfiring” of SS, is not

present for the Chebyshev spectral identification as in that case the dominant columns of Φ_{ss} are accurate enough for $\hat{\theta}_{ss}$ to yield the minimum prediction error utilizing just them, with a little (or no) help from the redundant columns. Secondly, it is noted that, whether employing the SS algorithm or not, the proposed Chebyshev spectral identification significantly outperforms the finite difference identification in terms of accuracy.

Figure 5-7, which compares the percent estimate error (in 2-norm) versus the number of samples used for finite difference and Chebyshev spectral identifications, illustrates the superior accuracy of the Chebyshev spectral identification over its finite difference counterpart. In particular, it is noted that with the error axis linearly scaled (Figure 5-7, upper) literally no error is perceivable for the Chebyshev spectral identification *with* or *without* SS even with the smallest number of samples. It is only when the y-axis is scaled logarithmically (Figure 5-7, lower) that the (rather small) error incurred when using the Chebyshev spectral identification becomes visible. (Even with mere 60 samples and no SS, the estimate error, measured in 2-norm, is less than 1%.) Moreover, a dramatic decrease (a hundredfold, in the limit, to be more specific) in estimate error is noticed when the proposed SS algorithm is employed.

For the finite difference identification, however, the estimate error is unacceptably large if a “small” number of samples are used. Furthermore, employing the SS algorithm only aggravates the estimates – this is the backfiring phenomenon discussed earlier. It is noticed that as the number of samples increases the estimates based on the finite difference identification *with* SS approach those based on the same identification but *without* SS; the more samples we provide the less backfiring we can expect to occur for

the finite difference identification. This stems from the fact that with more samples the grid is finer and hence finite difference methods are capable of approximating the derivatives more accurately. Probably most importantly, we should draw attention to the point that the proposed Chebyshev spectral identification outdoes its finite difference counterpart even with less number of samples and no SS for the former.

Figure 5-7 Comparison of Parameter estimate error in reconstruction of Allen-Cahn equation using finite differences and spectral methods with and without system simplification. *Upper*: Linear scale; *Lower*: Logarithmic scale.

Figure 5-8 compares the prediction error between finite difference and Chebyshev spectral identifications. As seen in this figure, as a result of the way our SS algorithm is defined (loosely, eliminate the terms in the model structure which contribute insignificantly to reducing the prediction error) for both of the identifications, the prediction error deteriorates after applying the SS algorithm. However, the prediction error for Chebyshev spectral identification remains small even after this deterioration and but it goes awry for finite difference identification. This once again demonstrates the

power of the Chebyshev spectral identification when combined with the SS algorithm: unlike for its finite difference counterpart, the proposed SS algorithm manages to improve the estimates errors for the Chebyshev spectral identification without impairing the prediction errors.

Figure 5-8 Prediction error comparison between finite difference and Chebyshev spectral identifications. *Upper*: Linear scale; *Lower*: Logarithmic scale. Due to definition of ERR and its threshold, for both identifications prediction error deteriorates with application of SS algorithm. Deterioration is minimal for the Chebyshev spectral identification and drastic for finite difference identification.

As the last point we again mention that our SS algorithm is not “sensitive” to the choice of the ERR threshold as long as the threshold is set to a value in an appropriate interval. Figure 5-9 illustrates this point by plotting the relative error in the dominant coefficients (namely the coefficients of u , u_{xx} , and u^3) of the Allen-Cahn equation versus the ERR threshold based on $N=100$ number of samples and the state data picked at $t=35$. It is seen that the relative errors in the dominant coefficients are minimum (and practically constant) with the %ERR threshold in the intervals $(\sim 0.025, \sim 0.1)$ and

($\sim 0.025, \sim 0.275$) for the finite difference and Chebyshev spectral identifications respectively. The SS algorithm, and hence the identification, would only fail if the ERR threshold is set in excess of a maximum, 0.1% for finite difference and 0.275% for Chebyshev spectral.

Figure 5-9 Effect of ERR threshold on relative error in dominant coefficients, namely those of u, u_{xx} , and u^3 , of Allen-Cahn system for (a) Finite Difference and (b) Chebyshev spectral identifications (number of samples $N = 100$, state data picked at $t = 35$). Relative errors in dominant coefficients are smallest with % ERR threshold in ($\sim 0.025, \sim 0.1$) for (a) and ($\sim 0.025, \sim 0.275$) for (b). Identification process would only fail if threshold is set in excess of a maximum, 0.1% for (a) and 0.275% for (b). Note: errors in coefficients of u and u^3 are not exactly equal; squares and stars do not exactly coincide.

CONCLUSION

In this work, it was demonstrated that the well-known accuracy of spectral methods for direct problems carries over to inverse problems as well. To the best of our knowledge, spectral methods have not been applied to identification problems before.

For the vast area of spatiotemporal system identification, it was shown that a black box model class based on Volterra expansion is capable of providing the required structure to approach the true system and the associated model structure was developed in one dimensions.

For approximating the derivatives present in that model structure, trigonometric and algebraic spectral operators were presented and the accuracy of the corresponding approximations was discussed. Specifically, it was noted that, for both methods, the decay rates of both truncation and aliasing errors increase with increasing smoothness of the function being approximated. This was mentioned to be in contrast to finite difference methods that have fixed polynomial error decay rates, independent of the smoothness of the function they approximate.

For producing the aforementioned spectral operators in one dimension, two basically equivalent, but computationally different, methods of discrete expansion and interpolation were introduced and the operators were created using fast Fourier transform, recursion, and matrix multiplication. At last, round-off efficient ways were discussed to implement the operators in floating-point-arithmetic software.

Having taken care of spectral operators was not enough, as validating their results required a benchmark against methods that were tried and true. For that reason, high

order finite differences were introduced and symbolic and numerical algorithms for automatically generating explicit or implicit finite difference stencils on arbitrarily-spaced grids were presented.

With the derivative approximation tools ready, we presented the notion of parameter estimation based on least squares, first just pragmatically and later from the viewpoint of stochastic theory. Orthogonal decompositions were coupled with the concept of error reduction ratio to both solve the least squares problem and implement an effective system simplification algorithm working towards eliminating redundant parameters to leave a parsimonious model.

Finally several numerical experiments were used to address different aspects of spatiotemporal system identification and spectral methods side by side. Specifically, we showed that spectral identification (trigonometric or algebraic) yields much more accurate parameter estimates compared to finite difference identification. This higher accuracy was illustrated to be present both for the same number of samples and even for *less* number of samples. This is of great practical significance as with spectral identification, *less* number of sensors may be used to estimate the model parameters with the *same or even higher* accuracy than that provided by finite difference identification. On the other hand, attention was drawn to the fact that spectral methods cannot do magic; below a critical number of samples (which is dependent on the spectral method in use and the highest wavenumber present in the output) aliasing occurs and the identification will not be accurate. Finally, positive effects of applying the proposed system simplification algorithm to the spectral identifications as well as the side effects of the algorithm when applied to finite difference identifications were also addressed and it was noted that the successful

simplification requires the error reduction ratio threshold to be in a certain interval but, apart from that, the results are not sensitive to the exact value of that threshold.

REFERENCES

- [1] Gouesbet, G., S. Meunier-Guttin-Cluzel, and O. Minard, *Chaos and its reconstruction*. 2003, New York: Nova Science Publishers. viii, 320 p.
- [2] Farmer, J.D. and J.J. Sidorowich, *Predicting chaotic time series*. Physical Review Letters, 1987. **59**(8): p. 845.
- [3] Timmer, J., H. Rust, W. Horbelt, and H.U. Voss, *Parametric, nonparametric and parametric modelling of a chaotic circuit time series*. Physics Letters A, 2000. **274**(3-4): p. 123-134.
- [4] Yule, G.U., *On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers*. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1927. **226**: p. 267-298.
- [5] Ljung, L., *System identification : theory for the user*. 2nd ed. 1999, Upper Saddle River, N.J.: Prentice Hall. xxii, 609 p.
- [6] Ljung, L., *Perspectives on System Identification*. Proc. of IFAC Conference on System Identification, 2008.
- [7] Lee, L.H. and K. Poolla, *Identification of Linear Parameter-Varying Systems Using Nonlinear Programming*. Journal of Dynamic Systems, Measurement, and Control, 1999. **121**(1): p. 71-78.
- [8] Bamieh, B. and L. Giarré, *Identification of linear parameter varying models*. International Journal of Robust and Nonlinear Control, 2002. **12**(9): p. 841-853.
- [9] Roll, J., A. Bemporad, and L. Ljung, *Identification of piecewise affine systems via mixed-integer programming*. Automatica, 2004. **40**(1): p. 37-50.
- [10] Zhao, X. and V.Z. Marmarelis, *On the relation between continuous and discrete nonlinear parametric models*. Automatica, 1997. **33**(1): p. 81-84.
- [11] Trefethen, L.N. and D. Bau, *Numerical linear algebra*. 1997, Philadelphia: SIAM. xii, 361 p.
- [12] Moler, C.B., *Numerical computing with MATLAB*. 2004, Philadelphia: Society for Industrial and Applied Mathematics. xi, 336 p.
- [13] Gauss, C.F., *Theory of Motion of the Heavenly Bodies Moving About the Sun in Conic Sections: A Translation of Theoria Motus*. 1809, New York: Dover.
- [14] Ljung, L., *Asymptotic variance expressions for identified black-box transfer function models*. Automatic Control, IEEE Transactions on, 1985. **30**(9): p. 834-844.
- [15] Strang, G., *Introduction to Applied Mathematics*. 1986, Wellesley, Mass.: Wellesley-Cambridge Press. ix, 758 p.
- [16] Ljung, L. and T. Söderström, *Theory and practice of recursive identification*. MIT Press series in signal processing, optimization, and control ; 4. 1983, Cambridge, Mass.: MIT Press. xix, 529 p.
- [17] Aguirre, L.A., U.S. Freitas, C. Letellier, and J. Maquet, *Structure-selection techniques applied to continuous-time nonlinear models*. Physica D: Nonlinear Phenomena, 2001. **158**(1-4): p. 1-18.

-
- [18] Chen, S., S.A. Billings, and W. Luo, *Orthogonal least squares methods and their application to non-linear system identification*. International Journal of Control, 1989. **50**(5): p. 1873 - 1896.
 - [19] Lanczos, C., *Trigonometric interpolation of empirical and analytical functions*. Journal of Mathematical Physics, 1938. **17**: p. 123-199.
 - [20] Fornberg, B., *A practical guide to pseudospectral methods*. Cambridge monographs on applied and computational mathematics ; 1. 1996, Cambridge [England] ; New York: Cambridge University Press. x, 231 p.
 - [21] Canuto, C., *Spectral methods in fluid dynamics*. Corr. 2nd print. ed. Springer series in computational physics. 1988, Berlin ; New York: Springer-Verlag. xiv, 567 p.
 - [22] Canuto, C., *Spectral methods in fluid dynamics*. Springer series in computational physics. 1988, New York: Springer-Verlag. xiv, 557.
 - [23] Hesthaven, J., S. Gottlieb, and D. Gottlieb, *Spectral methods for time-dependent problems*. Cambridge monographs on applied and computational mathematics ; 21. 2007, Cambridge: Cambridge University Press. ix, 273.
 - [24] Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*. 2nd ed. 1992, Cambridge [Cambridgeshire]; New York: Cambridge University Press. xxvi, 994 p.
 - [25] Trefethen, L.N., *Spectral methods in MATLAB*. Software, environments, tools. 2000, Philadelphia, PA: Society for Industrial and Applied Mathematics. xvi, 165 p.
 - [26] Hesthaven, J., S. Gottlieb, and D. Gottlieb, *Spectral methods for time-dependent problems*. Cambridge monographs on applied and computational mathematics; 21. 2007, Cambridge: Cambridge University Press. ix, 273.
 - [27] Golub, G.H. and J.H. Welsch, *Calculation of Gauss Quadrature Rules*. Mathematics of Computation, 1969. **23**(106): p. 221-s10.
 - [28] Boyd, J.P., *Chebyshev & Fourier spectral methods*. Lecture notes in engineering; 49. 1989, Berlin; New York: Springer-Verlag. xvi, 798 p.
 - [29] Welfert, B.D., *Generation of Pseudospectral Differentiation Matrices I*. SIAM Journal on Numerical Analysis, 1997. **34**(4): p. 1640-1657.
 - [30] Trefethen, L.N. and M.R. Trummer, *An instability phenomenon in spectral methods*. SIAM Journal of Numerical Analysis, 1987. **24**(5): p. 1008-1023.
 - [31] Breuer, K.S. and R.M. Everson, *On the errors incurred calculating derivatives using Chebyshev polynomials*. Journal of Computational Physics, 1992. **99**(1): p. 56-67.
 - [32] Golub, G.H. and C.F. Van Loan, *Matrix computations*. 3rd ed. Johns Hopkins studies in the mathematical sciences. 1996, Baltimore: Johns Hopkins University Press. xxvii, 694 p.
 - [33] Baltensperger, R. and J.P. Berrut, *The errors in calculating the pseudospectral differentiation matrices for Cebyshev-Gauss-Lobatto points*. Computers & Mathematics with Applications, 1999. **37**(1): p. 41-48.
 - [34] Tang, T. and M.R. Trummer, *Boundary Layer Resolving Pseudospectral Methods for Singular Perturbation Problems*. SIAM Journal on Scientific Computing, 1996. **17**(2): p. 430-438.
 - [35] Kosloff, D. and H. Tal-Ezer, *A Modified Chebyshev Pseudospectral Method with an $O(N-1)$ Time Step Restriction*. Journal of Computational Physics, 1993. **104**(2): p. 457-469.

-
- [36] Acton, F.S., *Numerical methods that work*. 1990, Washington, D.C.: Mathematical Association of America. xx, 549 p.
 - [37] Higham, N.J., *The numerical stability of barycentric Lagrange interpolation*. IMA J Numer Anal, 2004. **24**(4): p. 547-556.
 - [38] Powell, M.J.D., *Approximation theory and methods*. 1981, Cambridge [Eng.] ; New York: Cambridge University Press. x, 339 p.
 - [39] Trefethen, L.N. and J.A.C. Weideman, *Two results on polynomial interpolation in equally spaced points*. Journal of Approximation Theory, 1991. **65**(3): p. 247-260.
 - [40] Vertesi, P., *Optimal Lebesgue Constant for Lagrange Interpolation*. SIAM Journal on Numerical Analysis, 1990. **27**(5): p. 1322-1331.
 - [41] Meinardus, G., *Approximation of functions: theory and numerical methods*. Springer tracts in natural philosophy ; v. 13. 1967, Berlin, New York [etc.]: Springer-Verlag. viii, 198 p.
 - [42] Iserles, A., *A first course in the numerical analysis of differential equations*. Cambridge texts in applied mathematics. 1996, Cambridge ; New York: Cambridge University Press. xvi, 378 p.
 - [43] Fornberg, B., *Calculation of Weights in Finite Difference Formulas*. SIAM Review, 1998. **40**(3): p. 685-691.
 - [44] Bender, C.M. and S.A. Orszag, *Advanced mathematical methods for scientists and engineers I : asymptotic methods and perturbation theory*. 1999, New York: Springer. xiv, 593 p.
 - [45] Fornberg, B., *Generation of Finite Difference Formulas on Arbitrarily Spaced Grids*. Mathematics of Computation, 1988. **51**: p. 699-706.
 - [46] Packard, N.H., J.P. Crutchfield, J.D. Farmer, and R.S. Shaw, *Geometry from a Time Series*. Physical Review Letters, 1980. **45**(9): p. 712.
 - [47] Takens, F., *Dynamical Systems and Turbulence*. Lecture Notes in Mathematics, ed. Rand, D.A. and Young, L.S. Vol. 898 1981, Berlin: Springer-Verlag. 366-381.
 - [48] Sauer, T., J.A. Yorke, and M. Casdagli, *Embedology*. Journal of Statistical Physics, 1991. **65**(3): p. 579-616.
 - [49] Cox, S.M. and P.C. Matthews, *Exponential Time Differencing for Stiff Systems*. Journal of Computational Physics, 2002. **176**(2): p. 430-455.
 - [50] Kuramoto, Y. and T. Tsuzuki, *Persistent Propagation of Concentration Waves in Dissipative Media Far from Thermal Equilibrium*. Progress of Theoretical Physics, 1976. **55**(2): p. 356-369.
 - [51] Kuramoto, Y. and T. Tsuzuki, *On the Formation of Dissipative Structures in Reaction-Diffusion Systems*. Progress of Theoretical Physics, 1975. **54**(3): p. 687-699.
 - [52] Sivashinsky, G.I., *Nonlinear analysis of hydrodynamic instability in laminar flames--I. Derivation of basic equations*. Acta Astronautica, 1977. **4**(11-12): p. 1177-1206.
 - [53] Sivashinsky, G.I., *On flame propagation under conditions of stoichiometry*. SIAM Journal of Applied Mathematics, 1980. **39**(1): p. 67-82.
 - [54] Hyman, J.M. and B. Nicolaenko, *The Kuramoto-Sivashinsky equation: A bridge between PDE'S and dynamical systems*. Physica D: Nonlinear Phenomena, 1986. **18**(1-3): p. 113-126.
 - [55] Nicolaenko, B., B. Scheurer, and R. Temam, *Some global dynamical properties of the Kuramoto-Sivashinsky equations: Nonlinear stability and attractors*. Physica D: Nonlinear Phenomena, 1985. **16**(2): p. 155-183.

- [56] Aceves, A., H. Adachihara, C. Jones, J.C. Lerman, D.W. McLaughlin, J.V. Moloney, and A.C. Newell, *Chaos and coherent structures in partial differential equations*. Physica D: Nonlinear Phenomena, 1986. **18**(1-3): p. 85-112.
- [57] Söderström, T. and P. Stoica, *System identification*. Prentice-Hall international series in systems and control engineering. 1989, New York: Prentice Hall. xxiv, 612.
- [58] Jack, X., *Front Propagation in Heterogeneous Media*. SIAM Review, 2000. **42**(2): p. 161-230.
- [59] Xin, J.X. and J. Zhu, *Quenching and propagation of bistable reaction-diffusion fronts in multidimensional periodic media*. Physica D: Nonlinear Phenomena, 1995. **81**(1-2): p. 94-110.

PUBLICATIONS ARISING FROM THIS WORK

Spatiotemporal System Identification on Non-periodic Domains with Chebyshev Spectral Operators and System Reduction Algorithms (with Daolin Xu). Submitted to **Chaos**

Spatiotemporal System Reconstruction using Fourier Spectral Operators and Structure Selection Techniques (with Daolin Xu). **Chaos**, 2008. **18**(4): p. 043122.

Spatiotemporal System Identification with Spectral Methods (with Daolin Xu). 3rd East Asia **SIAM** Conference, Xiamen, China