Quasiseparable Approach to Evaluate Cubic Splines via A Fast and Stable Algorithm

Dylan English, Shreya Mittal, Patrick Rupp, and Genesis Vargas
Sirani M. Perera; Faculty Adviser
Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL.

A∫PiRE Feb. 11, 2017

Introduction

Splines are polynomials between every pair of tabulated points for the given data set. The coefficients of the polynomials are non-locally determined, which is ideal to ensure overall smoothness in the interpolated function up to and n^{th} order of derivative. The important aspect of cubic splines is that the require continuity in order to find equations for the second derivatives of the function. When presented a plotted data set with a scattered appearance or significance variation that linear interpolation cannot provide an ideal function for it, a cubic spline can best fit the data set in this situation. **In order to solve for cubic splines, we have to consider a matrix equation Ty=x, factor T into LDU form, follow the quasi-separable structure of L^{-1} and U^{-1}, and factor quasiseparable matrices.**

The development of fast and efficient algorithms is crucial not only for computer scientists, but also for mathematicians and engineers as those algorithms lead to reduced the complexity. Apart from the said common interest, these professionals seek paths to construct novel models using existing data points. This leads numerical analysts to explore interpolation techniques for modeling phenomena.

We will state **a quasiseparable approach to evaluate cubic splines through the derivation and design of a fast and stable algorithm**. The derivation is carried via **sparse factorizations** of **the inverse of tridiagonal matrices** in connection to the **quasiseparable matrices**. New factorization leads to an alternative method to **solve the system of tridiagonal matrices opposed to the existing methods.** Due to the stability results, the new algorithm is much more favorable than the existing unstable Thomas Algorithm. The resulting algorithm has **the lowest computational complexity comparing to the existing well accepted Thomas Algorithm**. Applications for the algorithm would be realistic for situations in electrical engineering, systems engineering, sensor processing, and parallel processing.

Methodology

To solve coefficients of cubic splines problem, we start with the equation

Ty=x
where

$$T=\begin{bmatrix} a_1 & b_2 & & & & 0\\ c_2 & a_2 & b_3 & & & \\ & c_3 & . & & & \\ & & & . & & \\ & & & & a_{n-1} & b_n\\ 0 & & & & c_n & a_n \end{bmatrix} \quad x=\begin{bmatrix} x_1\\ x_2\\ .\\ .\\ x_{n-1}\\ x_n \end{bmatrix} \quad \mathrm{y}=\begin{bmatrix} y_1\\ y_2\\ .\\ .\\ y_{n-1}\\ y_n \end{bmatrix}$$

Next, we use Gaussian elimination to factor the tridiagonal matrix T into a lower bi-diagonal matrix; *L*, a diagonal matrix; *D*, and an upper bi-diagonal matrix; *U*.

$$\rho_1=a_1 \quad \rho_k=a_k-\frac{c_k b_k}{\rho_{k-1}} \quad l_k=\frac{c_{k+1}}{\rho_k} \quad u_k=\frac{b_{k+1}}{\rho_k}$$

$$L=\begin{bmatrix} 1 & & & & 0\\ l_1 & 1 & & & \\ & l_2 & . & & \\ & & . & & \\ & & & 1 & \\ 0 & & & l_{n-1} & 1 \end{bmatrix} D=\begin{bmatrix} \rho_1 & & & & 0\\ & \rho_2 & & & \\ & & . & & \\ & & & . & \\ & & & \rho_{n-1} & \\ 0 & & & & \rho_n \end{bmatrix} U=\begin{bmatrix} 1 & u_1 & & & 0\\ & 1 & u_2 & & \\ & & . & & \\ & & & . & \\ & & & 1 & u_{n-1}\\ 0 & & & & 1 \end{bmatrix}$$

$$U^{-1}=\begin{bmatrix} 1 & -u_1 & u_1u_2 & -u_1u_2u_3 & \dots & & (-1)^{n-1}u_1\dots u_{n-1}\\ & 1 & -u_2 & u_2u_3 & & & (-1)^{n-2}u_2\dots u_{n-1}\\ & & . & & & & .\\ & & & . & & & .\\ & & & & 1 & -u_{n-2} & u_{n-2}u_{n-1}\\ & & & & & 1 & -u_{n-1}\\ 0 & & & & & & 1 \end{bmatrix}$$

We then invert each bi-diagonal L and U matrices. The resulting matrices form lower and upper quasiseparable matrices. Finally, we factor the lower and upper quasi-separable matrices into the product of sparse matrices; G_k and Γ_k where *k = 1, ..., n-1* and *D.*

$$G_k=\left[\begin{array}{c|cc|c} I_{k-1} & & & \\ \hline & 1 & -u_k & \\ & 0 & 1 & \\ \hline & & & I_{n-k-1} \end{array}\right] \quad \Gamma_k=\left[\begin{array}{c|cc|c} I_{n-k-1} & & & \\ \hline & 1 & 0 & \\ & -l_{n-k} & 1 & \\ \hline & & & I_{k-1} \end{array}\right]$$

$$D^{-1}=\begin{bmatrix} 1/\rho_1 & & & & 0\\ & 1/\rho_2 & & & \\ & & . & & \\ & & & . & \\ & & & 1/\rho_{n-1} & \\ 0 & & & & 1/\rho_n \end{bmatrix}$$

$$y=G_1\dots G_{n-1}D^{-1}\Gamma_1\dots\Gamma_{n-1}x$$

Results

This project is designed to derive a fast and stable algorithm to solve a system of linear equations in connection to cubic splines. **The well accepted Thomas Algorithm has 8N flops** which is small compared to a standard matrix inversion algorithm. **Here, we propose a new algorithm with 5N-4 flops**. The proposed algorithm also leads to stability as shown in the following table, where the **machine precision is 2.23e-16**.

Size	Relative Error (random a, b, c)	Relative Error (a = 4 and b, c = 1)
40x40	1.40E-15	1.69E-15
80x80	8.55E-16	1.55E-15
120x120	3.97E-15	2.92E-15
160x160	6.78E-15	2.88E-15
200x200	3.33E-15	3.96E-15
240x240	2.63E-15	2.11E-15
280x280	9.31E-16	1.24E-15
320x320	6.88E-15	1.65E-15
360x360	6.66E-15	4.78E-15
400x400	8.97E-16	2.41E-16

Table displays the first column holding dimensions of tridiagonal matrices, the second column with the relative error populated by a random tridiagonal matrix, and the last column with the tridiagonal matrix in cubic splines having a=4 and b=c=1

Conclusion

The proposed derivation leads to

- **Fast *O(n)* Algorithm**
- **Sparse Factorization**
- **Recursive Algorithm**
- **Stable Algorithm**
- **Accurate Algorithm**

References

[1] T. Bella, Y. Eidelman, I. Gohberg, I. Koltracht, and V. Olshevsky, *A Fast Bjorck-Pereyra-type Algorithm for Solving Hessenberg-quasiseparable-Vandermonde Systems*, SIAM Journal on Matrix Analysis 31(2), 790-815, (2009).

[2] I. Gohberg, T. Kailath, V. Olshevsky, *Fast Gaussian Elimination With Partial Pivoting For Matrices With Displacement Structure*, Mathematics of Computation 64(212), 1557 – 1576, (1995).

[3] W. T. Lee, *Tridiagonal Matrices and Thomas Algorithm*, Scientific Computation, University of Limerick.

[4] V. Olshevsky, G. Strang, P. Zhlobich, *Green's Matrices*, Linear Algebra and its Applications 432(1), 218-241, (2010).