Bernstein type inequalities of the Voronoi cell of the hexagonal lattice

Charles Ouyang

REU 2012

Fourier Analysis on the Circle

Definition

A family of kernels $\{K_n\}_{n=1}^{\infty}$ is said to be a family of good kernels if it satisfies the following properties:

(i) $\frac{1}{2\pi}\int_{-\pi}^{\pi} K_n(x)\,\mathrm{d}x = 1\ \forall n \in \mathbb{N}$

(ii) $\exists\, M > 0$ such that $\forall n \in \mathbb{N}\ \int_{-\pi}^{\pi} |K_n(x)|\,\mathrm{d}x \leq M$

(iii) $\forall\, \delta > 0\ \int_{\delta \leq |x| \leq \pi} |K_n(x)|\,\mathrm{d}x \to 0$ as $n \to \infty$

The Dirichlet Kernel

Definition

The n^{th} Dirichlet kernel $D_n(x)$ is given by $\sum_{k=-n}^{n} e^{ikx}$

Proposition

$$D_n(x) = \sum_{k=-n}^{n} e^{ikx} = 1 + \sum_{k=1}^{n} \cos(kx) = \frac{\sin((n+\frac{1}{2})x)}{\sin(x+\frac{1}{2})}$$

Proposition

The Dirichlet kernel is not a good kernel.

Fourier Series and Fourier Coefficients

Definition

The n^{th} Fourier coefficient $\hat{f}(n)$ is given by the integral

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-inx}\mathrm{d}x$$

Definition

The n^{th}-partial sum of the Fourier series of f is given by

$$S_N(f)(x) = \sum_{n=-N}^{N} \hat{f}(n)e^{inx}$$

Lemma

The n^{th}-partial sum can be written as a convolution of the original function with the n^{th} Dirichlet kernel. That is,

$$S_N(f)(x) = \sum_{n=-N}^{N} \hat{f}(n)e^{inx}$$

The Fejér kernel

Definition

We define the n^{th} Fejér kernel to be

$$F_n(x) = \frac{1}{n} \sum_{k=0}^{n-1} D_k(x)$$

Proposition

The Fejér kernel is a good kernel.

Approximation to the identity

Theorem

Let $\{K_n\}_{n=1}^{\infty}$ be a family of good kernels, and f an integrable function on the circle. Then

$$\lim_{n\to\infty}(f * K_n)(x) = f(x)$$

whenever f is continuous at x. If f is continuous everywhere, then the above limit is uniform.

Approximation to the identity

Proof.

Let x be a point of continuity of f. Consider

$$\begin{aligned}
|(f * K_n)(x) - f(x)| &= |\frac{1}{2\pi}\int_{-\pi}^{\pi} K_n(y)\, f(x-y)\,\mathrm{d}y - f(x)| \\
&= |\frac{1}{2\pi}\int_{-\pi}^{\pi} K_n(y)[f(x-y) - f(x)]\,\mathrm{d}y| \\
&\leq \frac{1}{2\pi}\int_{-\pi}^{\pi} |K_n(y)[f(x-y) - f(x)]|\,\mathrm{d}y \\
&\leq \frac{1}{2\pi}\int_{|y|<\delta} |K_n(y) f(x-y|)\,\mathrm{d}y + \frac{1}{2\pi}\int_{|y|\geq\delta} |K_n(y \\
&\leq \frac{M\epsilon}{2\pi} + \frac{B}{2\pi}\int_{|y|\geq\delta} K_n(y)\mathrm{d}y
\end{aligned}$$

$\square$

Numerical Approach for General NEP

Numerical Approach for General NEP

- Essential tools (Schur, Generalized Schur) not available
- Sensitivity analysis, Round-off error analysis under study

Numerical Approach for General NEP

- Essential tools (Schur, Generalized Schur) not available
- Sensitivity analysis, Round-off error analysis under study
- MATLAB can solve polynomial eigenvalue problems

Numerical Approach for General NEP

- Essential tools (Schur, Generalized Schur) not available
- Sensitivity analysis, Round-off error analysis under study
- MATLAB can solve polynomial eigenvalue problems
- Common approach for polynomial EP is **Linearization:**

$$\begin{bmatrix} A(\lambda) & 0 \\ 0 & I \end{bmatrix} = E(\lambda)(B - \lambda C)F(\lambda).$$

$$\det E(\lambda) = K_1 \neq 0 \qquad\qquad \det F(\lambda) = K_2 \neq 0$$

Numerical Approach for General NEP

Example

$$A(\lambda)x = (M\lambda^2 + C\lambda + K)x = 0.$$

Let $u = \lambda x$. Then, $\lambda M u + C u + K x = 0.$

Thus,

$$\begin{bmatrix} O & I \\ -K & -C \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} - \lambda \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = 0.$$

Numerical Approach for General NEP

Division of Numerical Methods

- Linearization (polynomial, rational)

Numerical Approach for General NEP

Division of Numerical Methods

- Linearization (polynomial, rational)
- Treat it in its original form

Numerical Approach for General NEP

Division of Numerical Methods

- Linearization (polynomial, rational)
- Treat it in its original form
- Methods for dense, small problems

Numerical Approach for General NEP

Division of Numerical Methods

- Linearization (polynomial, rational)
- Treat it in its original form
- Methods for dense, small problems
- Methods for sparse large problems

Numerical Approach for General NEP

Division of Numerical Methods

- Linearization (polynomial, rational)
- Treat it in its original form
- Methods for dense, small problems
- Methods for sparse large problems
- Structure-preserving methods

LU Factorization

$$[A \quad b] = \left[\begin{array}{ccc|c} 1 & 4 & 7 & -9 \\ 2 & 5 & 8 & 2 \\ 3 & 6 & 10 & 7 \end{array}\right] \xrightarrow[R_3-3R_1]{R_2-2R_1} \left[\begin{array}{ccc|c} 1 & 4 & 7 & -9 \\ 0 & -3 & -6 & 20 \\ 0 & -6 & -11 & 34 \end{array}\right]$$

Similarly, we have

$$\left[\begin{array}{ccc|c} 1 & 4 & 7 & -9 \\ 0 & -3 & -6 & 20 \\ 0 & -6 & -11 & 34 \end{array}\right] \xrightarrow{R_3-2R_2} \left[\begin{array}{ccc|c} 1 & 4 & 7 & -9 \\ 0 & -3 & -6 & 20 \\ 0 & 0 & 1 & -6 \end{array}\right]$$

The new system obtained is triangular:

$$\begin{aligned} x + 4y + 7z &= -9 \\ -3y - 6z &= 20 \\ z &= -6 \end{aligned}$$

LU Factorization

In general, for an $n \times n$ matrix A, we have that

$$L_{n-1} \cdots L_2 L_1 A = U, \tag{3.1}$$

so that

$$A = LU, \qquad \text{with} \qquad L = (L_{n-1} \cdots L_2 L_1)^{-1}.$$

If some pivots are zero, then $PA = LU$.

RRLU Factorization

General A

$$
\begin{aligned}
P_1 A P_2 &= LU \\
&= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}
\end{aligned}
$$

Rank Deficient A

$$
\begin{aligned}
P_1 A P_2 &= LU \\
&= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

RRLU Factorization of a Matrix Function

Theorem

Let $A(\lambda) \in C^2$ be a $n \times n$ matrix such that $A(\lambda_0)$ has a column rank of $n - m$, $m \leq n - 1$. Assume there are permutation matrices P_1, P_2 such that $P_1 A(\lambda_0) P_2 = L_0 U_0$, where L_0 is a block unit lower triangular matrix and U_0 is a block upper triangular matrix. Then, there is a neighborhood $N(\lambda_0)$ such that

$$P_1 A(\lambda) P_2 = L(\lambda) U(\lambda), \quad \forall\, \lambda \in N(\lambda_0),$$

with $L(\lambda_0) = L_0$, $U(\lambda_0) = U_0$; $L(\lambda)$ a block unit lower triangular matrix, $U(\lambda)$ a block upper triangular matrix, with $U_{22}(\lambda)$ differentiable at $\lambda = \lambda_0$.

Main Idea

Let λ^* be an eigenvalue of $A(\lambda)_{n\times n}$ with rank $A(\lambda^*) = n - m$

$$P_1^* A(\lambda^*) P_2^* = L(\lambda^*) U(\lambda^*),$$

where

$$U(\lambda^*) = \begin{bmatrix} U_{11}(\lambda^*) & U_{12}(\lambda^*) \\ 0 & 0 \end{bmatrix}.$$

Let λ_0 be close to λ^*, and $P_1^0 A(\lambda_0) P_2^0 = L(\lambda_0) U(\lambda_0)$, with

$$U(\lambda_0) = \begin{bmatrix} U_{11}(\lambda_0) & U_{12}(\lambda_0) \\ 0 & U_{22}(\lambda_0) \end{bmatrix}, \quad \|U_{22}\|_F \ll \|U_{11} \quad U_{12}\|_F.$$

From the theorem above,

$$P_1 A(\lambda) P_2 = L(\lambda) U(\lambda), \quad \forall \lambda \in N(\lambda_0).$$

Determining Nonlinear Eigenvalues Through Minimization

Goal

- Improve λ_0 as an approximation to λ^*.

Determining Nonlinear Eigenvalues Through Minimization

Goal

- Improve λ_0 as an approximation to λ^*.

Plan

- Give initial guess λ_0
- Perform rank revealing decomposition
- Minimize lower right block: $\|U_{22}(\lambda)\|_F \to 0$.
- Repeat until convergence

Minimization Technique

$$||U_{22}(\lambda)||_F^2 \approx ||U_{22}(\lambda_0) + U'_{22}(\lambda_0)(\lambda - \lambda_0)||_F^2$$

Choose next iterate λ_1 so that

$$||U_{2,2}(\lambda_0)+U'_{2,2}(\lambda_0)(\lambda_1-\lambda_0)||_F^2 = \min_{\lambda} ||U_{2,2}(\lambda_0)+U'_{2,2}(\lambda_0)(\lambda-\lambda_0)||_F^2.$$

Minimization using Newton's method

Letting $f(\lambda) = ||U_{2,2}(\lambda_0) + U'_{2,2}(\lambda_0)(\lambda - \lambda_0)||_F^2$,

Iterate:

$$\lambda_{i+1} = \lambda_i - \frac{f'(\lambda_i)}{f''(\lambda_i)},$$

where

$$f'(\lambda_i) = 2(\mathrm{col}\ U'_{2,2}(\lambda_i))^H \cdot \mathrm{col}\ U_{2,2}(\lambda_i),$$

$$f''(\lambda_i) = 2||U'_{2,2}(\lambda_i)||_F^2.$$

Algorithm for Computation of Nonlinear Eigenvalues

Step 1: Given an initial approximation λ_0 to λ_*
Step 2: Compute

$$A(\lambda_i) \text{ and } A'(\lambda_i),\ i = 0, 1, \cdots$$

Step 3: Compute the LU decomposition with complete pivoting of $A(\lambda_i)$:

$$P_1 A(\lambda_i) P_2 = L(\lambda_i) U(\lambda_i)$$

Algorithm to compute NE: Newton's method + RRLU

Step 4: Compute

$$U'_{2,2}(\lambda_i) = (L_i^{-1}P_1A'(\lambda_i)P_2)_{2,2} - (L_i^{-1}P_1A'(\lambda_i)P_2)_{2,1}({U_{1,1}^{(i)}}^{-1}U_{1,2}^{(i)})$$

Step 5: Compute

$$\lambda_{i+1} = \lambda_i - \frac{(\text{col } U'_{2,2}(\lambda_i))^H \cdot \text{col } U_{2,2}(\lambda_i)}{||U'_{2,2}(\lambda_i)||_F^2}.$$

Step 6: If tolerance is satisfied, stop. Otherwise, repeat steps 2-6.

Convergence is quadratic

Numerical Rank Determination

Property

Let $P_1^k A(\lambda^{(k)}) P_2^k = L(\lambda^{(k)}) U(\lambda^{(k)})$ Then, the diagonals of U satisfy

$$\min_{1 \le i \le n-m} |\, u_{ii}(\lambda^{(k)}) \,| \gg \max_{n-m+1 \le i,j \le n} |\, u_{ij}(\lambda^{(k)}) \,|$$

Numerical Rank Determination

Property

Let $P_1^k A(\lambda^{(k)}) P_2^k = L(\lambda^{(k)}) U(\lambda^{(k)})$ Then, the diagonals of U satisfy

$$\min_{1 \leq i \leq n-m} |\, u_{ii}(\lambda^{(k)}) \,| \gg \max_{n-m+1 \leq i,j \leq n} |\, u_{ij}(\lambda^{(k)}) \,|$$

Choose threshold $\epsilon > 0$ so that

$$\max_{n-m+1 \leq i,j \leq n} |u_{ij}(\lambda^{(k)})| \leq \epsilon \max_{1 \leq i \leq n-m} |u_{ii}(\lambda^{(k)})| \leq \min_{1 \leq i \leq n-m} |u_{ii}(\lambda^{(k)})|$$

100 x 100 Time Comparison

Table: Time [ms] Comparison of Algorithm Performance

Nonlinear Matrix	LU Average	QR Average	Ratio of Averages (QR / LU)
Q	391.154	1696.066	4.336
Q, E	362.494	1630.445	4.498
Q, S	393.234	1634.839	4.157
Q, E, S	389.039	1650.813	4.243

$$A(\lambda) = A_0 + A_1\lambda + A_2\lambda^2 + A_3\sin(\lambda) + A_4\cos(\lambda) + A_5 e^{\lambda}.$$

A_i are random constant matrices.

Newton Steffensen Method

Cubic Convergence Iterative Formula

Applying Steffensen's acceleration method to Newton's root finding method generates an iterative formula with cubic convergence. Let $f(x_*) = 0$ and let x_0 be sufficiently close to x_*, then the successive iterative approximations are determined by

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f'(x_n)(f(x_n) - f(x_n^*))}$$

where

$$x_n^* = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Newton Steffensen Method

$$x_{n+1} = x_n - \frac{f'^2(x_n)}{f''(x_n)(f'(x_n) - f'(x_n^*))}$$

where

$$x_n^* = x_n - \frac{f'(x_n)}{f''(x_n)}.$$

- $f'(\lambda) = (\mathrm{col}\ U'_{2,2}(\lambda_i))^H \cdot \mathrm{col}\ U_{2,2}(\lambda_i)$
- $f''(\lambda) = ||U'_{2,2}(\lambda_i)||_F^2$

Table: Cubic Algorithm Error Convergence

Iteration	Error
1	0.3212
2	0.1340
3	$1.7791 * 10^{-4}$
4	$5.1172 * 10^{-13}$

Some References

[1] T. Betcke, N. J. Higham, V. Mehrmann, C. Schrder and F. Tisseur *"NLEVP: A collection of nonlinear eigenvalue problems"* MIMS Eprints **40** (2008) .

[2] F. Tisseur and K. Meerbergen, *"The Quadratic Eigenvalue Problem"*. SIAM Review, **43** (2001) 235-286

[3] E.K. Chu, T.M. Hwang, W.W. Ling, C.T. Wu, *"Vibration of fast trains, palindromic eigenvalue problems and structure preserving doubling algorithms"*. J. Computational and Applied Math. **219** (2008) 237–252.