FORMALIZING COURANT-FISCHER IN LEAN

RUOCHENG WANG AND MIKHAIL ZAYTSEV

Abstract. We formalize the Courant–Fischer theorem for real symmetric matrices in Lean using the `mathlib` library. Both the max–min and min–max characterizations of eigenvalues are stated and proved in detail, with an emphasis on formulations that are amenable to formalization. Along the way, we discuss key design choices and technical challenges encountered in translating the classical proof into Lean.

1. Introduction

The Courant–Fischer theorem is a classical result in linear algebra and spectral theory, providing a variational characterization of the eigenvalues of a real symmetric matrix in terms of Rayleigh quotients. It plays a fundamental role in matrix analysis and has numerous applications, for example in numerical linear algebra, optimization, and spectral graph theory.

In this project, we formalize a version of the Courant–Fischer theorem for real symmetric matrices in the Lean theorem prover using the `mathlib` library. Our development includes both the max–min and min–max formulations of the theorem, together with the auxiliary results needed to make these statements well-defined in Lean. Rather than aiming for a library-ready contribution, our focus is on producing a complete and explicit formal proof that closely follows standard textbook arguments while remaining compatible with Lean's type system.

A central challenge of the formalization is that classical proofs of the Courant–Fischer theorem rely on informal arguments about extrema, dimensions of subspaces, and bilinearity of inner products, all of which require careful handling in Lean. In particular, the use of supremum and infimum requires explicit proofs of non-emptiness and boundedness of the relevant sets, and the computational expansion of inner products leads to lengthy but, to our knowledge, unavoidable calculations in the formal setting.

The paper is organized as follows. In Section 2, we review the necessary mathematical background and state the main auxiliary results used in the proof, emphasizing formulations that are suitable for formalization. The statements and proofs are partially based on [Sud25], but are adapted to better align with the requirements of Lean. In Section 3, we discuss the Lean implementation in detail, highlighting key design choices, technical obstacles, and points where the formal proof diverges from or elaborates on standard textbook presentations.

2. Mathematical Background

2.1. **The Basics.** Throughout this section, we will assume that $n \in \mathbb{N}$ is a natural number and $A \in \mathbb{R}^{(n+1)\times(n+1)}$ is a *real symmetric* matrix with eigenvalues $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $v_0, \ldots, v_n$.

First, we will show the following four computational results needed in the proof of our main theorem, which admit direct, though verbose, translations into Lean.

Lemma 2.1. *Let $I \subseteq \{0, \ldots, n\}$ be finite and $x = \sum_{i\in I} c_i v_i \in \mathbb{R}^{n+1}$ with $c_i \in \mathbb{R}$ for all $i \in I$. Then,*

$$\langle x, x\rangle = \sum_{i\in I} c_i^2.$$

Proof. The proof is a direct computation. Observe that

$$\begin{aligned}\langle x, x\rangle &= \left\langle \sum_{i\in I} c_i v_i, \sum_{j\in I} c_j v_j \right\rangle \\ &= \sum_{i\in I}\sum_{j\in I} c_i c_j \langle v_i, v_j\rangle \\ &= \sum_{i\in I} c_i^2,\end{aligned}$$

where the last equality follows from the orthonormality of $v_0, \ldots, v_n$. □

Lemma 2.2. *Let $I \subseteq \{0, \ldots, n\}$ be finite and $x = \sum_{i\in I} c_i v_i \in \mathbb{R}^{n+1}$ with $c_i \in \mathbb{R}$ for all $i \in I$. Then,*

$$\langle Ax, x\rangle = \sum_{i\in I} \lambda_i c_i^2.$$

Proof. The proof is similar to Lemma 2.1. We compute

$$\begin{aligned}\langle Ax, x\rangle &= \left\langle \sum_{i\in I} c_i (Av_i), \sum_{j\in I} c_j v_j \right\rangle \\ &= \left\langle \sum_{i\in I} \lambda_i c_i v_i, \sum_{j\in I} c_j v_j \right\rangle \\ &= \sum_{i\in I}\sum_{j\in I} \lambda_i c_i c_j \langle v_i, v_j\rangle \\ &= \sum_{i\in I} \lambda_i c_i^2,\end{aligned}$$

where we used that $Av_i = \lambda_i v_i$ for all i and the orthonormality of $v_0, \ldots, v_n$ to eliminate the off-diagonal terms. □

Remark 2.3. In Lean, the computations in the two proofs above are substantially more involved than their pen-and-paper counterparts. One must explicitly expand the bilinearity of the inner product, rewrite the action of A on eigenvectors, and use orthonormality to cancel all cross terms. Finding a formulation that is flexible enough to reuse these results required particular care.

Applying these two lemmas, we immediately obtain the following characterization.

Corollary 2.4. *Let $I \subseteq \{0, \dots, n\}$ be finite and $x = \sum_{i \in I} c_i v_i \in \mathbb{R}^{n+1}$ with $c_i \in \mathbb{R}$ nonzero for some $i \in I$. Then,*

$$\frac{\langle Ax, x \rangle}{\langle x, x \rangle} = \frac{\sum_{i \in I} \lambda_i c_i^2}{\sum_{i \in I} c_i^2}.$$

From this, we deduce the next corollary.

Corollary 2.5. *Let $x = \sum_{i=0}^{n} c_i v_i \in \mathbb{R}^{n+1} \setminus \{0\}$ with $c_i \in \mathbb{R}$ nonzero for some $i = 0, \dots, n$. Then,*

$$\lambda_n \leq \frac{\langle Ax, x \rangle}{\langle x, x \rangle} \leq \lambda_0.$$

Proof. The proof is again a direct computation. On the one hand,

$$\begin{aligned} \frac{\langle Ax, x \rangle}{\langle x, x \rangle} &= \frac{\sum_{i=0}^{n} \lambda_i c_i^2}{\sum_{i=0}^{n} c_i^2} \\ &\geq \frac{\sum_{i=0}^{n} \lambda_n c_i^2}{\sum_{i=0}^{n} c_i^2} = \lambda_n. \end{aligned}$$

On the other hand,

$$\begin{aligned} \frac{\langle Ax, x \rangle}{\langle x, x \rangle} &= \frac{\sum_{i=0}^{n} \lambda_i c_i^2}{\sum_{i=0}^{n} c_i^2} \\ &\leq \frac{\sum_{i=0}^{n} \lambda_0 c_i^2}{\sum_{i=0}^{n} c_i^2} = \lambda_0. \end{aligned}$$

□

We are now ready to define the main objects that we will investigate.

Definition 2.6 (Rayleigh-Set)**.** Let $S \leq \mathbb{R}^{n+1}$ be a subspace. We define the *Rayleigh-Set* of a matrix A and S by

$$\mathcal{R}(A, S) := \left\{ s \in \mathbb{R} : \exists x \in S \setminus \{0\} \left(s = \frac{\langle Ax, x \rangle}{\langle x, x \rangle} \right) \right\}.$$

Definition 2.7 (Courant-Fischer-Sets)**.** Let $k = 0, \dots, n$. We define the *Courant-Fischer-Sets* of a matrix A and k by:

$$\mathcal{F}(A, k) := \{t \in \mathbb{R} : \exists S \leq \mathbb{R}^{n+1} (\dim(S) = k + 1 \ \wedge \ t = \inf(\mathcal{R}(A, S)))\}$$

and

$\mathcal{C}(A,k) := \{t \in \mathbb{R} : \exists S \leq \mathbb{R}^{n+1}(\dim(S) = n-k+1 \ \land \ t = \sup(\mathcal{R}(A,S)))\}.$

Unfolding the definitions, the Courant–Fischer sets admit the following variational characterizations:

$$\sup(\mathcal{F}(A,k)) = \sup_{\substack{S \leq \mathbb{R}^{n+1} \\ \dim(S)=k+1}} \inf_{\substack{x \in S \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle},$$

and

$$\inf(\mathcal{C}(A,k)) = \inf_{\substack{S \leq \mathbb{R}^{n+1} \\ \dim(S)=n-k+1}} \sup_{\substack{x \in S \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle}.$$

These expressions correspond to the classical max–min and min–max formulations of the Courant–Fischer theorem, which we will establish in the next section.

In Lean, however, the operators sup and inf are only defined for sets that are known to be nonempty and suitably bounded. Consequently, before the variational statements above can be stated or proved, one must explicitly show that the Rayleigh set $\mathcal{R}(A,S)$ and the Courant–Fischer sets $\mathcal{F}(A,k)$ and $\mathcal{C}(A,k)$ satisfy these conditions. We therefore need to prove the following three propositions.

Proposition 2.8. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a matrix and let $S \leq \mathbb{R}^{n+1}$ be a subspace. Then the Rayleigh set $\mathcal{R}(A,S)$ is nonempty and bounded.*

Proof. The set $\mathcal{R}(A,S)$ is nonempty since S contains a nonzero vector whenever $\dim(S) > 0$, and any such vector defines a Rayleigh quotient.

Let $x \in S \setminus \{0\}$. Since the eigenvectors $v_0, \dots, v_n$ form an orthonormal basis of $\mathbb{R}^{n+1}$, we may write

$$x = \sum_{i=0}^{n} c_i v_i, \qquad \text{with } \sum_{i=0}^{n} c_i^2 > 0.$$

By Corollary 2.5, it immediately follows that $\mathcal{R}(A,S)$ is bounded. □

Proposition 2.9. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a matrix and let $k = 0, \dots, n$. Then the Courant–Fischer set $\mathcal{F}(A,k)$ is nonempty and bounded from above.*

Proof. The set $\mathcal{F}(A,k)$ is nonempty since the Rayleigh set $\mathcal{R}(A,S)$ is also nonempty for every subspace $S \leq \mathbb{R}^{n+1}$ of positive dimension.

Let $t \in \mathcal{F}(A,k)$. By definition, there exists a subspace $S \leq \mathbb{R}^{n+1}$ with $\dim(S) = k+1$ such that

$$t = \inf\big(\mathcal{R}(A,S)\big).$$

However, since $\mathcal{R}(A,S)$ is bounded, the claim on $\mathcal{F}(A,k)$ follows. □

Proposition 2.10. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a matrix and let $k = 0, \dots, n$. Then the Courant–Fischer set $\mathcal{C}(A,k)$ is nonempty and bounded from below.*

Proof. As before, the set $\mathcal{C}(A,k)$ is nonempty since the Rayleigh set $\mathcal{R}(A,S)$ is nonempty for every subspace $S \leq \mathbb{R}^{n+1}$ of positive dimension.

Let $t \in \mathcal{C}(A,k)$. By definition, there exists a subspace $S \leq \mathbb{R}^{n+1}$ with $\dim(S) = n-k+1$ such that

$$t = \sup\big(\mathcal{R}(A,S)\big).$$

However, since $\mathcal{R}(A,S)$ is bounded, the claim on $\mathcal{C}(A,k)$ follows. □

2.2. **Main Theorem.** We are now ready to state and prove our first version of the Courant–Fischer theorem.

Theorem 2.11. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a matrix with eigenvalues $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_n$ and corresponding orthonormal eigenvectors $v_0, \ldots, v_n$. Then, for any $k = 0, \ldots, n$,*

$$\lambda_k = \sup(\mathcal{F}(A,k)).$$

Proof. We will show two inequalities separately.

($\leq$). First, let $U = \operatorname{span}\{v_0, \ldots, v_k\}$. The vectors $v_0, \ldots, v_k$ are orthonormal and, in particular, linearly independent. Therefore, $\dim(U) = |\{v_0, \ldots, v_k\}| = k+1$.

Furthermore, by definition of U, we may write any nonzero $x \in U$ as follows: $x = \sum_{i=0}^{k} c_i v_i$ with $c_i \neq 0$ for some i.

We are now ready to apply Corollary 2.4 to the set $I = \{0, \ldots, k\}$ to obtain:

$$\frac{\langle Ax, x\rangle}{\langle x, x\rangle} = \frac{\sum_{i=0}^{k} \lambda_i c_i^2}{\sum_{i=0}^{k} c_i^2} \geq \frac{\sum_{i=0}^{k} \lambda_k c_i^2}{\sum_{i=0}^{k} c_i^2} = \lambda_k.$$

However, because $x \in U$ was an arbitrary nonzero element, we may conclude:

$$\inf_{\substack{x \in U \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \geq \lambda_k.$$

Finally, since we chose a particular $k+1$ dimensional subspace $U \leq \mathbb{R}^{n+1}$, we find:

$$\sup_{\substack{U \leq \mathbb{R}^{n+1} \\ \dim(U) = k+1}} \inf_{\substack{x \in U \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \geq \inf_{\substack{x \in U \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \geq \lambda_k,$$

which concludes the proof of the first inequality.

($\geq$). Now, let $U \leq \mathbb{R}^{n+1}$ be an arbitrary $k+1$ dimensional subspace of $\mathbb{R}^{n+1}$, define $W = \operatorname{span}\{v_k, \ldots, v_n\}$, and observe that:

$$\begin{aligned}\dim(U \cap W) &= \dim(U) + \dim(W) - \dim(U+W) \\ &\geq (k+1) + (n-k+1) - (n+1) = 1.\end{aligned}$$

Therefore, there exists a nonzero $x \in U \cap W$. In particular, $x \in W$, so we may write $x = \sum_{i=k}^{n} c_i v_i$ with $c_i \neq 0$ for some i.

Setting $I = \{k, \dots, n\}$, we are again in the right set-up to apply Corollary 2.4 to obtain:

$$\frac{\langle Ax, x\rangle}{\langle x, x\rangle} = \frac{\sum_{i=k}^{n} \lambda_i c_i^2}{\sum_{i=k}^{n} c_i^2} \le \frac{\sum_{i=k}^{n} \lambda_k c_i^2}{\sum_{i=k}^{n} c_i^2} = \lambda_k.$$

But, because $x \in U \cap W$, we also have $x \in U$. Thus, we get:

$$\inf_{\substack{x \in U \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \le \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \le \lambda_k.$$

And because U was arbitrary, we conclude:

$$\sup_{\substack{U \le \mathbb{R}^{n+1} \\ \dim(U) = k+1}} \inf_{\substack{x \in U \\ x \neq 0}} \frac{\langle Ax, x\rangle}{\langle x, x\rangle} \le \lambda_k,$$

which is exactly what we wanted to show. □

Remark 2.12. In the Lean formalization, the dimension argument in the second part of the proof requires working with integer-valued ranks rather than natural numbers, which increases the technical complexity compared to the pen-and-paper proof.

Theorem 2.13. *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a matrix with eigenvalues $\lambda_0 \ge \lambda_1 \ge \dots \ge \lambda_n$ and corresponding orthonormal eigenvectors $v_0, \dots, v_n$. Then, for any $k = 0, \dots, n$,*

$$\lambda_k = \inf(\mathcal{C}(A, k)).$$

Proof. Notice that $-A$ has eigenvalues $-\lambda_n \ge -\lambda_{n-1} \ge \dots \ge -\lambda_0$ and corresponding orthonormal eigenvectors $v_n, \dots, v_0$. Thus, applying Theorem 2.11 to $-A$ yields

$$\lambda_k = -(-\lambda_k) = -\sup(\mathcal{F}(-A, n-k)) = -\sup(-\mathcal{C}(A, k)) = \inf(\mathcal{C}(A, k)).$$

□

Remark 2.14. In Theorems 2.11 and 2.13, we do not explicitly assume that the matrix A is symmetric. Instead, symmetry is encoded implicitly through the existence of an orthonormal eigenbasis with real eigenvalues ordered as $\lambda_0 \ge \lambda_1 \ge \dots \ge \lambda_n$. In the formal development, this separation allows us to state and prove the Courant–Fischer inequalities independently of the Spectral Theorem, which is applied only later to derive these assumptions from symmetry.

The next version follows from an application of the Spectral Theorem together with the previous results.

Theorem 2.15 (Courant–Fischer)**.** *Let $A \in \mathbb{R}^{(n+1)\times(n+1)}$ be a real symmetric matrix. Then there exist eigenvalues $\lambda_0 \ge \lambda_1 \ge \dots \ge \lambda_n$ of A and, for any $k = 0, \dots, n$,*

$$\lambda_k = \sup(\mathcal{F}(A, k)) = \inf(\mathcal{C}(A, k)).$$

3. Implementation Notes

3.1. **Key Design Choices.** We briefly summarize the main structural decisions that shaped our formalization.

We work throughout in the Euclidean space $\mathbb{R}^{n+1}$ equipped with the `dotProduct`, rather than in a general real inner product space. The proof of the Courant–Fischer theorem is highly computational, relying on explicit expansions of inner products, matrix–vector multiplication, and finite sums. In this context, our hope was that working with matrices and vectors would be more convenient than reasoning abstractly. In addition, many standard applications of the Courant–Fischer theorem, such as those arising in spectral graph theory, are naturally formulated in terms of finite-dimensional Euclidean spaces and concrete matrices, making this setting particularly well suited both for formalization and for later use.

Rather than defining the Courant–Fischer expressions directly as nested suprema and infima, we introduce intermediate objects: the Rayleigh set $\mathcal{R}(A, S)$ and the Courant–Fischer sets $\mathcal{F}(A, k)$ and $\mathcal{C}(A, k)$. Each of these is defined as a subset of $\mathbb{R}$. This choice aligns closely with Lean's treatment of sup and inf as operations on sets, and allows us to reason uniformly about boundedness, non-emptiness, and extremal values using existing order-theoretic infrastructure.

In the core Courant–Fischer inequalities, we do not assume that the matrix A is symmetric. Instead, we work under the explicit assumptions that A admits a full orthonormal eigenbasis with real eigenvalues indexed by $\{0, \ldots, n\}$ and listed in non-increasing order. This separation allows us to prove the variational characterizations independently of the Spectral Theorem. In the final theorem, symmetry is used to invoke the Spectral Theorem in `mathlib`, which provides an orthonormal eigenbasis and a complete family of real eigenvalues. Since these eigenvalues are not returned in any canonical order, an additional reindexing step is required to sort them into non-increasing order before applying the Courant–Fischer results.

3.2. **Limitations and Pain Points.** While (most of) the design choices described above were made deliberately, a substantial portion of the development was shaped by limitations of the current state of Lean and `mathlib`, rather than by mathematical necessity. We summarize the main sources of friction we encountered, all of which required nontrivial workarounds.

A recurring source of overhead in our formalization is the treatment of supremum and infimum. In Lean, these operations are only available once non-emptiness and boundedness of the underlying set have been established. In informal mathematics, such conditions are typically left implicit or verified once and then ignored, whereas in the formal setting they must be carried through every statement involving extremal values.

This issue becomes particularly visible when passing from the max–min to the min–max formulation by considering the matrix $-A$. On paper, the identity

$$\sup(-X) = -\inf(X)$$

for a bounded set $X \subset \mathbb{R}$ is completely routine. In Lean, however, even this step requires explicit proofs that the relevant sets are non-empty and bounded, as well as careful rewriting of the definitions. Since the Courant–Fischer sets $\mathcal{F}(A,k)$ and $\mathcal{C}(A,k)$ are defined indirectly via quantification over subspaces and extremal values of Rayleigh quotients, showing that they behave well under the transformation $A \mapsto -A$ is slightly more involved than the corresponding informal argument suggests.

One might hope to avoid some of this overhead by allowing supremum and infimum to take the values $\pm\infty$, as is common in analysis. While Lean does support ordered types with adjoined infinities, such as the extended real numbers, this approach does not interact smoothly with the surrounding linear algebra infrastructure in `mathlib`. Adopting it would therefore shift complexity elsewhere rather than remove it. As a result, we chose to work entirely in $\mathbb{R}$ and accept the need for explicit boundedness and non-emptiness proofs.

Another major difficulty stems from Lean's strict separation between closely related numeric types. Dimensions of submodules are expressed in $\mathbb{N}$, while standard dimension arguments naturally involve subtraction and are more comfortably carried out in $\mathbb{Z}$. As a result, even simple textbook arguments such as

$$\dim(U \cap W) = \dim(U) + \dim(W) - \dim(U + W)$$

required repeated coercions between $\mathbb{N}$ and $\mathbb{Z}$. Though we managed to partially resolve this issue, arguments related to dimensions ended up being longer than expected.

From a user perspective, it is not always clear why such conversions require so much explicit work, given that the intended mathematical meaning is unambiguous. Smoother interaction between $\mathbb{N}$ and $\mathbb{Z}$ in common algebraic contexts would substantially improve usability.

A further source of technical overhead comes from manipulating finite sums and algebraic expressions involving inner products. Conceptually, Lemmas 2.1 and 2.2 are elementary computations: one expands a double sum using bilinearity, applies orthonormality, and cancels all off-diagonal terms. In Lean, however, such calculations require explicit control over every rewrite, sum expansion, and index alignment. As a result, even straightforward algebraic manipulations become long and fragile, and they constitute a significant portion of the overall code.

Similar issues arise when working with sums indexed by subsets. Mathematically, it is natural to sum over subsets of $\{0, \dots, n\}$. In Lean, however, moving between sets, subtypes, and finite index types introduces substantial type-level overhead. Summing over subtypes often led to type mismatches that made otherwise natural lemmas difficult to reuse in later arguments.

To mitigate this, we reformulated key lemmas using arbitrary finite index types together with injective maps into `Fin (n+1)`. This allowed us to control the indexing explicitly and avoid repeated coercions. While this approach is formally robust and reusable, it feels like an overly technical workaround. We did not find a significantly cleaner way to carry out these computations within the current framework. Although there is a possibility that sticking to the inner product machinery in general inner product spaces could have been helpful in this regard.

4. Outlook and Further Directions

The present formalization was designed to be complete and explicit, but it is not yet fully written in native `mathlib` style. A natural next step would be to refactor parts of the development with greater use of existing abstractions for finite-dimensional linear algebra, orthonormal families, and finite sums. Such a refactoring could reduce code size and computations, as well as improve readability, but it would require a careful reworking of many technical lemmas and is beyond the scope of this project.

Another natural direction is to extend the result to self-adjoint operators on finite-dimensional complex Hilbert spaces. While the high-level structure of the argument remains the same, formalizing this generalization in Lean would require additional work to adapt the definitions of Rayleigh quotients and to interface with the complex inner product space infrastructure in `mathlib`.

Finally, the Courant–Fischer theorem serves as a foundation for several further results in spectral theory, e.g., Cauchy's Interlacing Theorem. With the variational characterizations in place, formalizing such results would be a natural continuation of this work, as well as a step toward applications in areas such as spectral graph theory.

References

[Sud25] Benny Sudakov. Algebraic methods in combinatorics. Lecture notes, 2025.