



Math 673

Multigrid Methods: A Mostly Matrix-Based Approach

Chapter 02: The Two-Grid Algorithm

Abner J. Salgado and Steven M. Wise

asalgad1@utk.edu swise1@utk.edu
University of Tennessee

Fall 2024



Chapter 02

The Two-Grid Algorithm

Overview



In this chapter, we give the basics of the two-grid algorithm. Multigrid is basically a recursive form of the two-grid algorithm. Thus, understanding the two-grid algorithm is almost essential for understanding multigrid methods.

Our approach is based on matrices rather than operators. This helps the reader focus on implementation issues and dimensional considerations. This chapter will present the material in an otherwise generic framework. There is no connection to elliptic PDE's yet. But, be patient, and we will get there.

We will build the two-grid algorithm upon the GLIS framework that we discussed in the last chapter. Though it will not be obvious at first glance, we will show that, remarkably, the two-grid algorithm is a more complicated example of a particular GLIS.



Components of the Two-Grid Algorithm



Definition (Two-Grid Algorithm Components)

Suppose $n_0, n_1 \in \mathbb{N}$ and $n_1 \geq n_0$. Suppose $A_1 \in \mathbb{R}^{n_1 \times n_1}$ is SPD. A_1 is called the **fine grid stiffness matrix**. Choose a matrix $R_0 \in \mathbb{R}^{n_0 \times n_1}$, such that R_0 has full rank, that is, $\text{rank}(R_0) = n_0$. R_0 is called the **restriction matrix**. Define

$$P_0 := R_0^T \in \mathbb{R}^{n_1 \times n_0}.$$

P_0 is called the **prolongation matrix**. Choose a matrix $A_0 \in \mathbb{R}^{n_0 \times n_0}$ that is SPD. A_0 is called the **coarse grid stiffness matrix**. We say that the coarse-grid stiffness matrix, A_0 , **satisfies the Galerkin condition** iff it is defined via

$$A_0 := \underbrace{R_0}_{n_0 \times n_1} \underbrace{A_1}_{n_1 \times n_1} \underbrace{R_0^T}_{n_1 \times n_0} \in \mathbb{R}^{n_0 \times n_0}. \quad (1)$$



Visualization of the Galerkin Condition

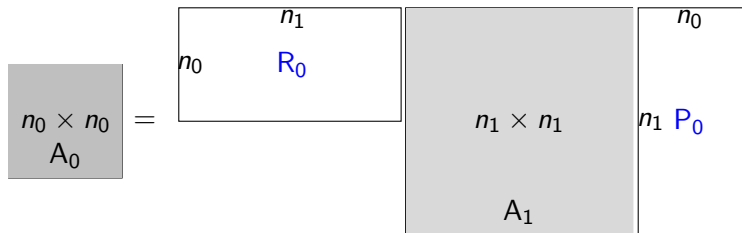


Figure: The creation of the coarse grid matrix via the Galerkin condition.



Proposition

If $A_1 \in \mathbb{R}^{n_1 \times n_1}$ is SPD and A_0 satisfies the Galerkin condition, then $A_0 \in \mathbb{R}^{n_0 \times n_0}$ is also SPD.

Proof.

(Symmetry-ness): Since $A_0 = R_0 A_1 R_0^T$ and A_1 is SPD, then

$$A_0^T = R_0 A_1^T R_0^T = R_0 A_1 R_0^T = A_0.$$

(Positive definite-ness): Let $\mathbf{w} \in \mathbb{R}^{n_0}$ an arbitrary vector and $\mathbf{w} \neq \mathbf{0}$. Since R_0 has full rank, then $R_0^T \mathbf{w} \neq \mathbf{0}$. Moreover, since A_1 is SPD,

$$\mathbf{w}^T A_0 \mathbf{w} = \mathbf{w}^T R_0 A_1 R_0^T \mathbf{w} = \left(R_0^T \mathbf{w} \right)^T A_1 \left(R_0^T \mathbf{w} \right) > 0.$$





Remark

Observe the actions of R_0 and P_0 :

$$R_0 \in \mathbb{R}^{n_0 \times n_1}, \quad R_0 \mathbf{v}_1 \in \mathbb{R}^{n_0}, \quad \forall \mathbf{v}_1 \in \mathbb{R}^{n_1},$$

and

$$P_0 \in \mathbb{R}^{n_1 \times n_0}, \quad P_0 \mathbf{v}_0 \in \mathbb{R}^{n_1}, \quad \forall \mathbf{v}_0 \in \mathbb{R}^{n_0}.$$

In other words, R_0 sends a vector from a higher-dimensional space into a lower-dimensional space. R_0 destroys information. P_0 sends a vector from a lower-dimensional space into a higher-dimensional space. P_0 creates information. See figure on the next page.

Information Lost in Restriction

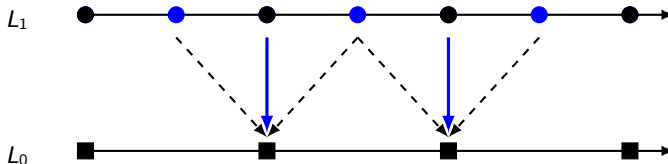


Figure: Restriction from a 1D fine grid (level 1, or L_1) to a 1D coarse grid (level 0, or L_0). Information is lost through restriction, since there are fewer degrees of freedom left in the coarse grid. Restriction can be thought of as a kind of data compression.

Information Created in Prolongation

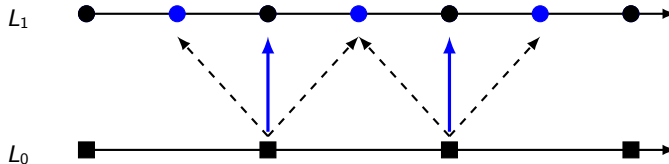


Figure: Prolongation from a 1D coarse grid (level 0, or L_0) to a 1D fine grid (level 1, or L_1). Information is created through prolongation, usually through some process of interpolation.



We wish to solve the following: given $\mathbf{f}_1 \in \mathbb{R}^{n_1}$ and $A_1 \in \mathbb{R}^{n_1 \times n_1}$ (SPD), find $\mathbf{u}_1^E \in \mathbb{R}^{n_1}$ such that

$$A_1 \mathbf{u}_1^E = \mathbf{f}_1.$$

To do this, we define a sequence of approximations

$$\mathbf{u}_1^{k+1} := \text{TG}(\mathbf{f}_1, \mathbf{u}_1^k)$$

using the two-grid operator, which we now define.



Definition (Two-Grid Operator)

Suppose that $\mathbf{g}_1, \mathbf{u}_1^{(0)} \in \mathbb{R}^{n_1}$ are given. The vector $\mathbf{u}_1^{(3)} \in \mathbb{R}^{n_1}$ is computed via the **two-grid operator**

$$\mathbf{u}_1^{(3)} := \text{TG} \left(\mathbf{g}_1, \mathbf{u}_1^{(0)} \right),$$

as follows:

① Pre-smoothing:

- $\mathbf{u}_1^{(1,0)} := \mathbf{u}_1^{(0)};$
- $\mathbf{u}_1^{(1,\sigma+1)} := \mathbf{u}_1^{(1,\sigma)} + S_1 \left(\mathbf{g}_1 - A_1 \mathbf{u}_1^{(1,\sigma)} \right), \quad 1 \leq \sigma \leq m_1 - 1;$
- $\mathbf{u}_1^{(1)} := \mathbf{u}_1^{(1,m_1)}.$



Definition (Two-Grid Operator, Cont.)

Suppose that $\mathbf{g}_1, \mathbf{u}_1^{(0)} \in \mathbb{R}^{n_1}$ are given. The vector $\mathbf{u}_1^{(3)} \in \mathbb{R}^{n_1}$ is computed via the **two-grid operator**

$$\mathbf{u}_1^{(3)} := \text{TG} \left(\mathbf{g}_1, \mathbf{u}_1^{(0)} \right),$$

as follows:

② Coarse grid correction:

- $\mathbf{r}_1^{(1)} := \mathbf{g}_1 - \mathbf{A}_1 \mathbf{u}_1^{(1)};$
- $\mathbf{r}_0^{(1)} := \mathbf{R}_0 \mathbf{r}_1^{(1)};$
- $\mathbf{q}_0^{(1,E)} := \mathbf{A}_0^{-1} \mathbf{r}_0^{(1)};$
- $\mathbf{q}_0^{(1)} = \mathbf{q}_0^{(1,E)};$
- $\mathbf{q}_1^{(1)} := \mathbf{P}_0 \mathbf{q}_0^{(1)};$
- $\mathbf{u}_1^{(2)} := \mathbf{u}_1^{(1)} + \mathbf{q}_1^{(1)}.$



Definition (Two-Grid Operator, Cont.)

Suppose that $\mathbf{g}_1, \mathbf{u}_1^{(0)} \in \mathbb{R}^{n_1}$ are given. The vector $\mathbf{u}_1^{(3)} \in \mathbb{R}^{n_1}$ is computed via the **two-grid operator**

$$\mathbf{u}_1^{(3)} := \text{TG} \left(\mathbf{g}_1, \mathbf{u}_1^{(0)} \right),$$

as follows:

③ Post-smoothing:

- $\mathbf{u}_1^{(3,0)} := \mathbf{u}_1^{(2)};$
- $\mathbf{u}_1^{(3,\sigma+1)} := \mathbf{u}_1^{(3,\sigma)} + S_1^T (\mathbf{g}_1 - A_1 \mathbf{u}_1^{(3,\sigma)}), \quad 0 \leq \sigma \leq m_2 - 1;$
- $\mathbf{u}_1^{(3)} := \mathbf{u}_1^{(3,m_2)}.$

The element $\mathbf{q}_0^{(1)} \in \mathbb{R}^{n_0}$ is called the **two-grid coarse grid correction**. The element $\mathbf{q}_0^{(1,E)} \in \mathbb{R}^{n_0}$ is called the **exact coarse grid correction**. The matrix S_1 is called the **smoothing operator** or **smoothing matrix**.



Remark

In the two-grid setting, $\mathbf{q}_0^{(1)} = \mathbf{q}_0^{(1,E)}$. This is a very important feature of the two-grid algorithm. In the multigrid algorithm, these two vectors will not be the same, and much of our analysis will focus on estimating their difference.

Remark

*The element $\mathbf{q}_1^{(1)} \in \mathbb{R}^{n_1}$ is also often referred to as the **coarse grid correction**, even though it lives in level 1 (the fine level). Perhaps $\mathbf{q}_1^{(1)}$ would be more appropriately called the **fine grid correction**, but this name has not stuck. In the finite element context, $\mathbf{q}_0^{(1)} \in \mathbb{R}^{n_0}$ and $\mathbf{q}_1^{(1)} \in \mathbb{R}^{n_1}$ are typically different basis coordinates for the same function, the finite element coarse grid correction function.*

Remark

Notice that we have pre-symmetrized the smoothing operations by using the iterator S_1^T in the post smoothing step. These smoothing operations are classical GLIS's that we introduced in the last chapter.



Definition

Let m_1 and m_2 be nonnegative integers. Suppose that $\mathbf{u}_1^k \in \mathbb{R}^{n_1}$ is given. Then the iteration process

$$\mathbf{u}_1^{k+1} := \text{TG}(\mathbf{f}_1, \mathbf{u}_1^k)$$

defines the **generic two-grid algorithm** for solving

$$A_1 \mathbf{u}_1^E = \mathbf{f}_1.$$

The two-grid algorithm is called **one-sided** iff $m_1 \geq 1$ and $m_2 = 0$. The algorithm is called **symmetric** iff $m_1 = m_2$.



Remark

Let us examine the coarse-grid correction in a very special case. Suppose that A_0 satisfies the Galerkin condition, $n_0 = n_1$, and

$$R_0 = I_1, \quad (n_1 \times n_1 \text{ identity matrix}).$$

Then

$$A_0 = R_0 A_1 P_0 = R_0 A_1 R_0^T = A_1.$$

Let us compute the coarse grid correction:

$$\mathbf{r}_1^{(1)} := \mathbf{f}_1 - A_1 \mathbf{u}_1^{(1)};$$

$$\mathbf{r}_0^{(1)} := R_0 \mathbf{r}_1^{(1)} = \mathbf{r}_1^{(1)};$$

$$\mathbf{q}_0^{(1)} := A_0^{-1} \mathbf{r}_0^{(1)} = A_1^{-1} \mathbf{r}_1^{(1)} = \mathbf{e}_1^{(1)}.$$



Remark (Cont.)

Then,

$$\begin{aligned}
 \mathbf{u}_1^{(2)} &:= \mathbf{u}_1^{(1)} + P_0 \mathbf{q}_0^{(1)} \\
 &= \mathbf{u}_1^{(1)} + P_0 \mathbf{e}_1^{(1)} \\
 &= \mathbf{u}_1^{(1)} + \mathbf{e}_1^{(1)} \\
 &= \mathbf{u}_1^{(1)} + \mathbf{u}_1^E - \mathbf{u}_1^{(1)} \\
 &= \mathbf{u}_1^E,
 \end{aligned}$$

where

$$A_1 \mathbf{u}_1^E = \mathbf{f}_1.$$



Remark (Cont.)

Of course, the two-grid algorithm should terminate at this stage, because we have the exact solution. Now, in general

$$\mathbf{q}_0^{(1)} \neq \mathbf{e}_1^{(1)}.$$

But, the approximation

$$\mathbf{P}_0 \mathbf{q}_0^{(1)} \approx \mathbf{e}_1^{(1)}.$$

can be a quite good, in some sense. We will revisit this issue later.



The Coarse-Grid Ritz Projection



Now, let us show that the two-grid algorithm is a GLIS. We need only derive the iterator matrix, or, equivalently, the error propagation matrix. We need the following definitions first.

Definition (Coarse-Grid Ritz Projection)

Referring to a previous definition, we define the matrix

$$\tilde{\Pi}_1 = R_0^T \Pi_0 \in \mathbb{R}^{n_1 \times n_1},$$

where

$$\Pi_0 = \underbrace{A_0^{-1}}_{n_0 \times n_0} \underbrace{R_0}_{n_0 \times n_1} \underbrace{A_1}_{n_1 \times n_1} \in \mathbb{R}^{n_0 \times n_1}.$$

$\tilde{\Pi}_1$ is called the **coarse-grid Ritz projection matrix**.



Proposition

The matrix $\tilde{\Pi}_1$, as defined previously, is a bona fide projection matrix, that is,

$$\tilde{\Pi}_1 \tilde{\Pi}_1 = \tilde{\Pi}_1,$$

provided A_0 satisfies the Galerkin condition.

Proof.

Using the Galerkin condition (1),

$$\begin{aligned} \tilde{\Pi}_1 \tilde{\Pi}_1 &= R_0^T \Pi_0 R_0^T \Pi_0 \\ &= R_0^T A_0^{-1} \underbrace{R_0 A_1 R_0^T}_{A_0} A_0^{-1} R_0 A_1 \\ &= R_0^T A_0^{-1} R_0 A_1 \\ &= R_0^T \Pi_0 \\ &= \tilde{\Pi}_1. \end{aligned}$$





Corollary

If the Galerkin Condition holds, $I_1 - \tilde{\Pi}_1$ is also a projection matrix, that is,

$$(I_1 - \tilde{\Pi}_1) (I_1 - \tilde{\Pi}_1) = I_1 - \tilde{\Pi}_1,$$



The smoothing operation is just the application of a GLIS a certain number of times. We define the error transfer for this process in the expected way.

Definition (Smoothing Error Transfer Matrix)

Define the **pre-smoothing error transfer matrix** as

$$K_1 := I_1 - S_1 A_1.$$



Definition (Adjoint)

Let $M \in \mathbb{R}^{n_1 \times n_1}$. Define the **level-1 inner product** via

$$(u_1, v_1)_1 := u_1^T v_1, \quad \forall u_1, v_1 \in \mathbb{R}^{n_1}.$$

The **level-1 adjoint** of M is the unique matrix M^T that satisfies

$$(Mu_1, v_1)_1 = (u_1, M^T v_1)_1, \quad \forall u_1, v_1 \in \mathbb{R}^{n_1}.$$

(Of course, M^T is the usual transpose of M .) We say that M is **level-1 self-adjoint**, or **level-1 symmetric**, iff $M = M^T$. When the context is clear, we will drop the 'level-1' part of the terms above.



Definition (Adjoint, Cont.)

Define the A_1 **inner product** via

$$(\mathbf{u}_1, \mathbf{v}_1)_{A_1} := (A_1 \mathbf{u}_1, \mathbf{v}_1)_1, \quad \forall \mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{n_1}.$$

The A_1 **adjoint of** M is the unique matrix M^* satisfying

$$(M \mathbf{u}_1, \mathbf{v}_1)_{A_1} = (\mathbf{u}_1, M^* \mathbf{v}_1)_{A_1}, \quad \forall \mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{n_1}.$$

We say that M is A_1 **self-adjoint**, or A_1 **symmetric**, iff $M = M^*$. We say that M is A_1 **symmetric positive definite** iff it is A_1 symmetric and

$$(M \mathbf{u}_1, \mathbf{u}_1)_{A_1} > 0, \quad \forall \mathbf{u}_1 \in \mathbb{R}_*^{n_1}.$$



Proposition

Suppose that K_1 is the pre-smoothing error transfer matrix. Then,

$$K_1^* = I_1 - S_1^T A_1.$$



Proof.

To see this, let $\mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{n_1}$ be arbitrary. Then

$$\begin{aligned}
 ((I_1 - S_1 A_1) \mathbf{u}_1, \mathbf{v}_1)_{A_1} &= (A_1 (I_1 - S_1 A_1) \mathbf{u}_1, \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, (I_1 - S_1 A_1)^T A_1^T \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, (I_1 - A_1^T S_1^T) A_1 \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, (I_1 - A_1 S_1^T) A_1 \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, (A_1 - A_1 S_1^T A_1) \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, A_1 (I_1 - S_1^T A_1) \mathbf{v}_1)_1 \\
 &= (A_1 \mathbf{u}_1, (I_1 - S_1^T A_1) \mathbf{v}_1)_1 \\
 &= (\mathbf{u}_1, (I_1 - S_1^T A_1) \mathbf{v}_1)_{A_1}.
 \end{aligned}$$





The Two-Grid Error Transfer Matrix



Theorem

For the two-grid algorithm defined earlier, we have

$$\mathbf{e}_1^{k+1} = E_1 \mathbf{e}_1^k,$$

where

$$E_1 = (K_1^*)^{m_2} (I_1 - \tilde{\Pi}_1) (K_1)^{m_1},$$

and

$$\mathbf{e}_1^k := \mathbf{u}_1^E - \mathbf{u}_1^k$$

where \mathbf{u}_1^E is the exact solution, that is,

$$A_1 \mathbf{u}_1^E = \mathbf{f}_1.$$

Furthermore, the two-grid algorithm is a GLIS, that is, there is some $B_1 \in \mathbb{R}^{n_1 \times n_1}$ such that

$$E_1 = I_1 - B_1 A_1.$$



Proof.

Set

$$\mathbf{e}_1^{(i)} := \mathbf{u}_1^E - \mathbf{u}_1^{(i)}, \quad i = 1, 2, 3.$$

Then, the error after the pre-smoothing step is

$$\mathbf{e}_1^{(1)} := (\mathbf{I}_1 - \mathbf{S}_1 \mathbf{A}_1)^{m_1} \mathbf{e}_1^k = \mathbf{K}_1^{m_1} \mathbf{e}_1^k.$$

Next we have the coarse-grid correction step.

$$\begin{aligned} \mathbf{e}_1^{(2)} &= \mathbf{u}_1^E - \mathbf{u}_1^{(2)} \\ &= \mathbf{u}_1^E - \mathbf{u}_1^{(1)} - \mathbf{R}_0^T \mathbf{q}_0^{(1)} \\ &= \mathbf{e}_1^{(1)} - \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{r}_0^{(1)} \\ &= \mathbf{e}_1^{(1)} - \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \mathbf{r}_1^{(1)} \\ &= \mathbf{e}_1^{(1)} - \mathbf{R}_0^T \mathbf{A}_0^{-1} \mathbf{R}_0 \mathbf{A}_1 \mathbf{e}_1^{(1)} \\ &= \left(\mathbf{I}_1 - \tilde{\mathbf{\Pi}}_1 \right) \mathbf{e}_1^{(1)}. \end{aligned}$$



Proof (Cont.)

So

$$\mathbf{e}_1^{(2)} = \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right) \mathbf{K}_1^{m_1} \mathbf{e}_1^k.$$

Finally, after the post-smoothing step,

$$\mathbf{e}_1^{k+1} = \mathbf{e}_1^{(3)} = \left(\mathbf{I}_1 - \mathbf{S}_1^T \mathbf{A}_1 \right)^{m_2} \mathbf{e}_1^{(2)}.$$

So

$$\mathbf{e}_1^{k+1} = (\mathbf{K}_1^*)^{m_2} \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right) \mathbf{K}_1^{m_1} \mathbf{e}_1^k,$$

as desired.

We leave the second part as an exercise. □



Remark

In the case that $m_1 = m_2 = 1$, we have the following simple form for B_1 :

$$B_1 = \underbrace{S_1^T (S_1^{-T} + S_1^{-1} - A_1) S_1}_{\text{symmetric smoothing}} + \underbrace{K_1^* R_0^T A_0^{-1} R_0 K_1^{*T}}_{\text{coarse grid correction}},$$

which is, of course, symmetric. In general, we can show that B_1 is symmetric in the Euclidean sense iff $m_1 = m_2 = m$.



Theorem

The error propagation matrix for the two-grid algorithm, E_1 , is symmetric with respect to $(\cdot, \cdot)_{A_1}$ iff $m_1 = m_2 = m$. Furthermore, if $m_1 = m_2 = m$ and, if the Galerkin condition holds for A_0 , then E_1 is symmetric positive semi-definite (SPSD) with respect to $(\cdot, \cdot)_{A_1}$.



Proof.

(Symmetry): Let $\mathbf{u}_1, \mathbf{v}_1 \in \mathbb{R}^{n_1}$ be arbitrary. Then we have

$$\begin{aligned} (\mathbf{E}_1 \mathbf{u}_1, \mathbf{v}_1)_{A_1} &= \left((\mathbf{K}_1^*)^{m_2} \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right) \mathbf{K}_1^{m_1} \mathbf{u}_1, \mathbf{v}_1 \right)_{A_1} \\ &= \left(\left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right) \mathbf{K}_1^{m_1} \mathbf{u}_1, \mathbf{K}_1^{m_2} \mathbf{v}_1 \right)_{A_1} \\ &= \left(\mathbf{K}_1^{m_1} \mathbf{u}_1, \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right)^* \mathbf{K}_1^{m_2} \mathbf{v}_1 \right)_{A_1} \\ &= \left(\mathbf{u}_1, (\mathbf{K}_1^*)^{m_1} \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right)^* \mathbf{K}_1^{m_2} \mathbf{v}_1 \right)_{A_1}. \end{aligned}$$

Observe that

$$\left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right)^* = \left(\mathbf{I}_1 - \tilde{\mathbf{P}}_1 \right).$$



Proof (Cont.)

Indeed,

$$\left((I_1 - \tilde{\Pi}_1) \mathbf{u}_1, \mathbf{v}_1 \right)_{A_1} = (\mathbf{u}_1, \mathbf{v}_1)_{A_1} - \left(\tilde{\Pi}_1 \mathbf{u}_1, \mathbf{v}_1 \right)_{A_1}$$

and

$$\begin{aligned} \left(\tilde{\Pi}_1 \mathbf{u}_1, \mathbf{v}_1 \right)_{A_1} &= \left(A_1 R_0^T A_0^{-1} R_0 A_1 \mathbf{u}_1, \mathbf{v}_1 \right)_1 \\ &= \left(R_0^T A_0^{-1} R_0 A_1 \mathbf{u}_1, A_1 \mathbf{v}_1 \right)_1 \\ &= \left(A_1 \mathbf{u}_1, \left(R_0^T A_0^{-1} R_0 \right)^T A_1 \mathbf{v}_1 \right)_1 \\ &= \left(A_1 \mathbf{u}_1, R_0^T A_0^{-1} R_0 A_1 \mathbf{v}_1 \right)_1 \\ &= \left(A_1 \mathbf{u}_1, \tilde{\Pi}_1 \mathbf{v}_1 \right)_1 \\ &= \left(\mathbf{u}_1, \tilde{\Pi}_1 \mathbf{v}_1 \right)_{A_1}. \end{aligned}$$



Proof (Cont.)

Therefore

$$\tilde{\Pi}_1^* = \tilde{\Pi}_1,$$

and

$$\left(I_1 - \tilde{\Pi}_1 \right)^* = \left(I_1 - \tilde{\Pi}_1 \right).$$

The symmetry of E_1 follows iff $m_1 = m_2$.

(PSD-ness): Let $\mathbf{u}_1 \in \mathbb{R}^{n_1}$ be arbitrary and suppose $m_1 = m_2 = m$. Since the Galerkin condition holds, from a corollary, we have

$$\left(I_1 - \tilde{\Pi}_1 \right)^2 = \left(I_1 - \tilde{\Pi}_1 \right).$$



Proof (Cont.)

Now

$$\begin{aligned}
 (E_1 \mathbf{u}_1, \mathbf{u}_1)_{A_1} &= \left((K_1^*)^m (I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1, \mathbf{u}_1 \right)_{A_1} \\
 &= \left((I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1, K_1^m \mathbf{u}_1 \right)_{A_1} \\
 &= \left((I_1 - \tilde{P}_1)^2 K_1^m \mathbf{u}_1, K_1^m \mathbf{u}_1 \right)_{A_1} \\
 &= \left((I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1, (I_1 - \tilde{P}_1)^* K_1^m \mathbf{u}_1 \right)_{A_1} \\
 &= \left((I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1, (I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1 \right)_{A_1} \\
 &= \left\| (I_1 - \tilde{P}_1) K_1^m \mathbf{u}_1 \right\|_{A_1}^2 \geq 0.
 \end{aligned}$$

