9

MultiFreedom Constraints II

TABLE OF CONTENTS

§9.1. Introduction

In this Chapter we continue the discussion of methods to treat multifreedom constraints (MFCs). The master-slave method described previously was easy to explain, but exhibits serious shortcomings for treating arbitrary constraints (although the method has important applications to model reduction).

We now pass to the study of two other methods: penalty augmentation and Lagrange multiplier adjunction. Both techniques are better suited to general implementations of the Finite Element Method, whether linear or nonlinear.

§9.2. The Penalty Method

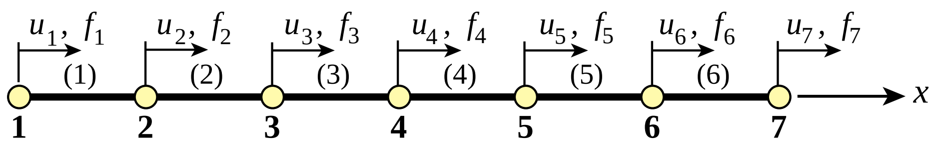

FIGURE 9.1. The example structure of Chapter 8, repeated for convenience.

§9.2.1. Physical Interpretation of Penalty Method

The penalty method will be first presented using a physical argument, leaving the mathematical formulation to a subsequent section. Consider again the example structure of Chapter 8, which is reproduced in Figure 9.1 for convenience. To impose $u_2 = u_6$ imagine that nodes 2 and 6 are connected with a "fat" bar of axial stiffness w, labeled with element number 7, as shown in Figure 9.2. This bar is called a *penalty element* and w is its *penalty weight*.

Such an element, albeit fictitious, can be treated exactly like another bar element insofar as continuing the assembly of the master stiffness equations. The penalty element stiffness equations, $\mathbf{K}^{(7)}\mathbf{u}^{(7)} = \mathbf{f}^{(7)}$, are[1]

$$w \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9.1}$$

Because there is one freedom per node, the two local element freedoms map into global freedoms 2 and 6, respectively. Using the assembly rules of Chapter 3 we obtain the following modified master stiffness equations: $\widehat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}}$, which shown in detail are

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22}+w & K_{23} & 0 & 0 & -w & 0 \\ 0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\ 0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\ 0 & -w & 0 & 0 & K_{56} & K_{66}+w & K_{67} \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}. \tag{9.2}$$

This system can now be submitted to the equation solver. Note that $\hat{\mathbf{u}} \equiv \mathbf{u}$, and only $\mathbf{K}$ has changed.

[1] The general method to construct these equations is described in §9.1.4.

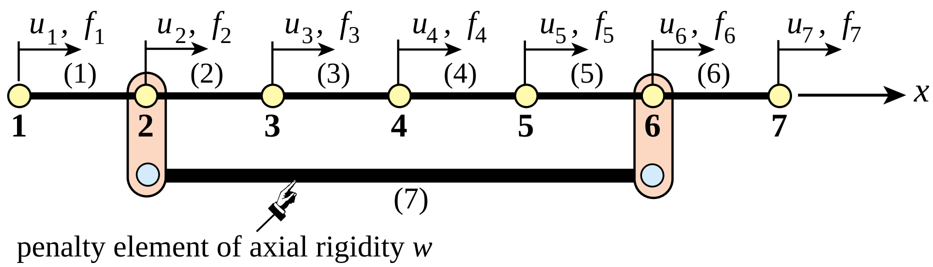

FIGURE 9.2. Adjunction of a fictitious penalty bar of axial stiffness w, identified as element 7,to enforce the MFC $u_2 = u_6$.

§9.2.2. Choosing the Penalty Weight

What happens when (9.2) is solved numerically? If a *finite* weight w is chosen the constraint $u_2 = u_6$ is approximately satisfied in the sense that one gets $u_2 - u_6 = e_g$, where $e_g \neq 0$. The "gap error" e_g is called the *constraint violation*. The magnitude $|e_g|$ of this violation depends on the weight: the larger w, the smaller the violation. More precisely, it can be shown that $|e_g|$ becomes proportional to $1/w$ as w gets to be sufficiently large (see Exercises). For example, raising w from, say, 10^6 to 10^7 can be expected to cut the constraint violation by roughly 10 if the physical stiffnesses are small compared to w.

Therefore it seems as if the proper strategy should be: try to make w as large as possible while respecting computer overflow limits. However, this is misleading. As the penalty weight w tends to ∞ the modified stiffness matrix in (9.2) becomes more and more *ill-conditioned with respect to inversion*.

To make this point clear, suppose for definiteness that the rigidities $E^e A^e / L^e$ of the actual bars $e = 1, \ldots 6$ are unity, that $w >> 1$, and that the computer solving the stiffness equations has a floating-point precision of 16 decimal places. Numerical analysts characterize such precision by saying that $\epsilon_f = O(10^{-16})$, where $|\epsilon_f|$ is the smallest power of 10 that perceptibly adds to 1 in floating-point arithmetic.[2] The modified stiffness matrix of (9.2) becomes

$$\widehat{\mathbf{K}} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2+w & -1 & 0 & 0 & -w & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & -w & 0 & 0 & -1 & 2+w & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{9.3}$$

As $w \to \infty$ rows 2 and 6, as well as columns 2 and 6, tend to become linearly dependent; in fact the negative of each other. But *linear dependency means singularity*. Therefore $\widehat{\mathbf{K}}$ approaches singularity as $w \to \infty$. In fact, if w exceeds $1/\epsilon_f = 10^{16}$ the computer will not be able to distinguish $\widehat{\mathbf{K}}$ from an exactly singular matrix. If $w << 10^{16}$ but $w >> 1$, the effect will be seen in increasing solution errors affecting the computed displacements $\hat{\mathbf{u}}$ returned by the equation solver. These errors, however, tend to be more of a random nature than the constraint violation error.

---

[2] Such definitions are more rigurously done by working with binary numbers and base-2 arithmetic but for the present discussion the use of decimal powers is sufficient.

§9.2.3. The Square Root Rule

Plainly we have two effects at odds with each other. Making w larger reduces the constraint violation error but increases the solution error. The best w is that which makes both errors roughly equal in absolute value. This tradeoff value is difficult to find aside of systematically running numerical experiments. In practice the heuristic *square root rule* is often followed.

This rule can be stated as follows. Suppose that the largest stiffness coefficient, before adding penalty elements, is of the order of 10^k and that the working machine precision is p digits.[3] Then choose penalty weights to be of order $10^{k+p/2}$ with the proviso that such a choice would not cause arithmetic overflow.[4]

For the above example in which $k \approx 0$ and $p \approx 16$, the optimal w given by this rule would be $w \approx 10^8$. This w would yield a constraint violation and a solution error of order 10^{-8}. Note that there is no simple way to do better than this accuracy aside from using extended (e.g., quad) floating-point precision. This is not easy to do when using standard low-level programming languages.

The name "square root" arises because the recommended w is in fact $10^k\sqrt{10^p}$. It is seen that picking the weight by this rule requires knowledge of both stiffness magnitudes and floating-point hardware properties of the computer used, as well as the precision selected by the program.

§9.2.4. Penalty Elements for General MFCs

For the constraint $u_2 = u_6$ the physical interpretation of the penalty element is clear. Nodal points 2 and 6 must move in lockstep long x, which can be approximately enforced by the heavy bar device shown in Figure 9.2. But how about $3u_3 + u_5 - 4u_6 = 1$? Or just $u_2 = -u_6$?

The treatment of more general constraints is linked to the theory of *Courant penalty functions*, which in turn is a topic in variational calculus. Because the necessary theory given in §9.1.5 is viewed as an advanced topic, the procedure used for constructing a penalty element is stated here as a recipe. Consider the homogeneous constraint

$$3u_3 + u_5 - 4u_6 = 0. \tag{9.4}$$

Rewrite this equation in matrix form

$$\begin{bmatrix} 3 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = 0, \tag{9.5}$$

and premultiply both sides by the transpose of the coefficient matrix:

$$\begin{bmatrix} 3 \\ 1 \\ -4 \end{bmatrix} \begin{bmatrix} 3 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 9 & 3 & -12 \\ 3 & 1 & -4 \\ -12 & -4 & 16 \end{bmatrix} \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = \bar{\mathbf{K}}^e \mathbf{u}^e = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{9.6}$$

---

[3] Such order-of-magnitude estimates can be readily found by scanning the diagonal of $\mathbf{K}$ because the largest stiffness coefficient of the actual structure is usually a diagonal entry.

[4] If overflows occurs, the master stiffness should be scaled throughout or a better choice of physical units made.

Here $\bar{\mathbf{K}}^e$ is the *unscaled* stiffness matrix of the penalty element. This is now multiplied by the penalty weight w and assembled into the master stiffness matrix following the usual rules. For the example problem, augmenting (9.2) with the w-scaled penalty element (9.6) yields

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\ 0 & K_{23} & K_{33}+9w & K_{34} & 3w & -12w & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\ 0 & 0 & 3w & K_{45} & K_{55}+w & K_{56}-4w & 0 \\ 0 & 0 & -12w & 0 & K_{56}-4w & K_{66}+16w & K_{67} \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}. \tag{9.7}$$

If the constraint is nonhomogeneous the force vector is also modified. To illustrate this effect, consider the MFC: $3u_3 + u_5 - 4u_6 = 1$. Rewrite in matrix form as

$$[\,3 \quad 1 \quad -4\,] \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = 1. \tag{9.8}$$

Premultiply both sides by the transpose of the coefficient matrix:

$$\begin{bmatrix} 9 & 3 & -12 \\ 3 & 1 & -4 \\ -12 & -4 & 16 \end{bmatrix} \begin{bmatrix} u_3 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -4 \end{bmatrix}. \tag{9.9}$$

Scaling by w and assembling yields

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\ 0 & K_{23} & K_{33}+9w & K_{34} & 3w & -12w & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\ 0 & 0 & 3w & K_{45} & K_{55}+w & K_{56}-4w & 0 \\ 0 & 0 & -12w & 0 & K_{56}-4w & K_{66}+16w & K_{67} \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3+3w \\ f_4 \\ f_5+w \\ f_6-4w \\ f_7 \end{bmatrix}. \tag{9.10}$$

§9.2.5. *The Theory Behind the Penalty Method

The rule comes from the following mathematical theory. Suppose we have a set of m linear MFCs. Using the matrix notation introduced in §8.1.3, these will be stated as

$$\mathbf{a}_p \mathbf{u} = b_p, \quad p = 1, \ldots m \tag{9.11}$$

where $\mathbf{u}$ contains all degrees of freedom and each $\mathbf{a}_p$ is a row vector with same length as $\mathbf{u}$. To incorporate the MFCs into the FEM model one selects a weight $w_p > 0$ for each constraints and constructs the so-called Courant quadratic penalty function or “penalty energy”

$$P = \sum_{p=1}^{m} P_p, \qquad \text{with} \quad P_p = \mathbf{u}^T \left(\tfrac{1}{2}\mathbf{a}_p^T \mathbf{a}_p \mathbf{u} - w_p \mathbf{a}_p^T b_p \right) = \tfrac{1}{2}\mathbf{u}^T \mathbf{K}^{(p)} \mathbf{u} - \mathbf{u}^T \mathbf{f}^{(p)}, \tag{9.12}$$

where we have called $\mathbf{K}^{(p)} = w_p \mathbf{a}_p^T \mathbf{a}_p$ and $\mathbf{f}^{(p)} = w_p \mathbf{a}^T b_i$. P is added to the potential energy function $\Pi = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f}$ to form the *augmented potential energy* $\Pi_a = \Pi + P$. Minimization of Π_a with respect to $\mathbf{u}$ yields

$$\Big(\mathbf{K}\mathbf{u} + \sum_{p=1}^{m} \mathbf{K}^{(p)}\Big)\,\mathbf{u} = \mathbf{f} + \sum_{p=1}^{m} \mathbf{f}^{(p)}. \tag{9.13}$$

Each term of the sum on p, which derives from term P_p in (9.12), may be viewed as contributed by a penalty element with globalized stiffness matrix $\mathbf{K}^{(p)} = w_p \mathbf{a}_p^T \mathbf{a}_p$ and globalized added force term $\mathbf{f}^{(p)} = w_p \mathbf{a}_p^T b_p$.

To use a even more compact form we may write the set of multifreedom constraints as $\mathbf{A}\mathbf{u} = \mathbf{b}$. Then the penalty augmented system can be written compactly as

$$(\mathbf{K} + \mathbf{A}^T\mathbf{W}\mathbf{A})\,\mathbf{u} = \mathbf{f} + \mathbf{W}\mathbf{A}^T\mathbf{b}, \tag{9.14}$$

where $\mathbf{W}$ is a diagonal matrix of penalty weights. This compact form, however, conceals the configuration of the penalty elements.

§9.2.6. Assessment of the Penalty Method

The main advantage of the penalty function method is its straightforward computer implementation. Looking at modified systems such as (9.2), (9.7) or (9.10) it is obvious that the master equations need not be rearranged. That is, $\mathbf{u}$ and $\hat{\mathbf{u}}$ are the same. Constraints may be programmed as "penalty elements," and stiffness and force contributions of these elements merged through the standard assembler. In fact using this method there is no need to distinguish between unconstrained and constrained equations! Once all elements — regular and penalty — are assembled, the system can be passed to the equation solver.[5]

An important advantage with respect to the master-slave (elimination) method is its lack of sensitivity with respect to whether constraints are linearly dependent. To give a simplistic example, suppose that the constraint $u_2 = u_6$ appears twice. Then two penalty elements connecting 2 and 6 will be inserted, doubling the intended weight but not otherwise causing undue harm.

An advantage with respect to the Lagrange multiplier method described in §9.2 is that positive definiteness is not lost. Such loss can affect the performance of certain numerical processes.[6] Finally, it is worth noting that the penalty method is easily extendible to nonlinear constraints although such extension falls outside the scope of this book.

The main disadvantage, however, is a serious one: the choice of weight values that balance solution accuracy with the violation of constraint conditions. For simple cases the square root rule previously described often works, although its effective use calls for knowledge of the magnitude of stiffness coefficients. Such knowledge may be difficult to extract from a general purpose "black box" program. For difficult cases selection of appropriate weights may require extensive numerical experimentation, wasting the user time with numerical games that have no bearing on the actual objective, which is getting a solution.

The deterioration of the condition number of the penalty-augmented stiffness matrix can have serious side effects in some solution procedures such as eigenvalue extraction or iterative solvers.

---

[5] Single freedom constraints, such as those encountered in Chapter 3, are usually processed separately for efficiency.

[6] For example, solving the master stiffness equations by Cholesky factorization or conjugate-gradients.

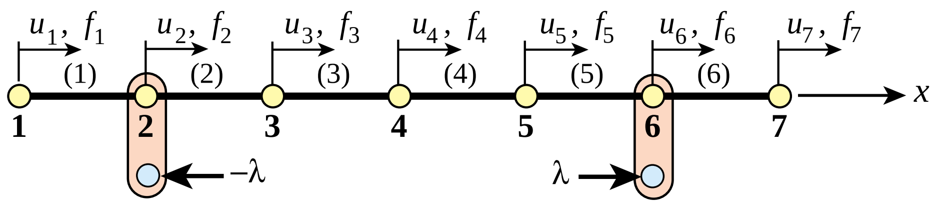

FIGURE 9.3. Physical interpretation of Lagrange multiplier adjunction to enforce the MFC $u_2 = u_6$.

Finally, even if optimal weights are selected, the combined solution error cannot be lowered beyond a threshold value.

From this assessment it is evident that penalty augmentation, although superior to the master-slave method from the standpoint of generality and ease of implementation, is no panacea.

§9.3. Lagrange Multiplier Adjunction

§9.3.1. Physical Interpretation

As in the case of the penalty function method, the method of Lagrange multipliers can be given a rigorous justification within the framework of variational calculus. But in the same spirit it will be introduced for the example structure from a physical standpoint that is particularly illuminating.

Consider again the constraint $u_2 = u_6$. Borrowing some ideas from the penalty method, imagine that nodes 2 and 6 are connected now by a *rigid* link rather than a flexible one. Thus the constraint is imposed exactly. But of course the penalty method with an infinite weight would "blow up."

We may remove the link if it is replaced by an appropriate reaction force pair $(-\lambda, +\lambda)$, as illustrated in Figure 9.3. These are called the *constraint forces*. Incorporating these forces into the original stiffness equations (8.10) we get

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\ 0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\ 0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\ 0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 - \lambda \\ f_3 \\ f_4 \\ f_5 \\ f_6 + \lambda \\ f_7 \end{bmatrix}. \tag{9.15}$$

This λ is called a *Lagrange multiplier*. Because λ is an unknown, let us transfer it to the *left hand side* by *appending* it to the vector of unknowns:

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 1 \\ 0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & -1 \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}. \tag{9.16}$$

But now we have 7 equations in 8 unknowns. To render the system determinate, the constraint condition $u_2 - u_6 = 0$ is appended as eighth equation:

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 1 \\ 0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & -1 \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ 0 \end{bmatrix}, \tag{9.17}$$

This is called the *multiplier-augmented* system. Its coefficient matrix, which is symmetric, is called the *bordered stiffness matrix*. The process by which λ is appended to the vector of original unknowns is called *adjunction*. Solving this system provides the desired solution for the degrees of freedom while also characterizing the constraint forces through λ.

§9.3.2. Lagrange Multipliers for General MFCs

The general procedure will be stated first as a recipe. Suppose that we want to solve the example structure subjected to three MFCs

$$u_2 - u_6 = 0, \qquad 5u_2 - 8u_7 = 3, \qquad 3u_3 + u_5 - 4u_6 = 1, \tag{9.18}$$

Adjoin these MFCs as the eighth, nineth and tenth equations:

$$\begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 \\ 0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 \\ 0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 \\ 0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 \\ 0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} \\ 0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & -8 \\ 0 & 0 & 3 & 0 & 1 & -4 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ 0 \\ 3 \\ 1 \end{bmatrix}, \tag{9.19}$$

Three Lagrange multipliers: λ_1, λ_2 and λ_3, are required to take care of three MFCs. Adjoin those unknowns to the nodal displacement vector. Symmetrize the coefficient matrix by appending 3 columns that are the transpose of the 3 last rows in (9.19), and filling the bottom right-hand corner with a 3×3 zero matrix:

$$
\begin{bmatrix}
K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{12} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 1 & 5 & 0 \\
0 & K_{23} & K_{33} & K_{34} & 0 & 0 & 0 & 0 & 0 & 3 \\
0 & 0 & K_{34} & K_{44} & K_{45} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & K_{45} & K_{55} & K_{56} & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & K_{56} & K_{66} & K_{67} & -1 & 0 & -4 \\
0 & 0 & 0 & 0 & 0 & K_{67} & K_{77} & 0 & -8 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 5 & 0 & 0 & 0 & 0 & -8 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 1 & -4 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}
=
\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ 0 \\ 3 \\ 1 \end{bmatrix}. \tag{9.20}
$$

§9.3.3. *The Theory Behind Lagrange Multipliers

The recipe illustrated by (9.20) comes from a well known technique of variational calculus. Using the matrix notation introduced in §8.1.3, compactly denote the set of m MFCs by $\mathbf{Au} = \mathbf{b}$, where $\mathbf{A}$ is $m \times n$. The potential energy of the unconstrained finite element model is $\Pi = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f}$. To impose the constraint, adjoin m Lagrange multipliers collected in vector $\boldsymbol{\lambda}$ and form the Lagrangian

$$
L(\mathbf{u}, \boldsymbol{\lambda}) = \Pi + \boldsymbol{\lambda}^T(\mathbf{Au} - \mathbf{b}) = \tfrac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} + \boldsymbol{\lambda}^T(\mathbf{Au} - \mathbf{b}). \tag{9.21}
$$

Extremizing L with respect to $\mathbf{u}$ and $\boldsymbol{\lambda}$ yields the multiplier-augmented form

$$
\begin{bmatrix} \mathbf{K} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}
\begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix}
=
\begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix}. \tag{9.22}
$$

The master stiffness matrix $\mathbf{K}$ in (9.22) is said to be *bordered* with $\mathbf{A}$ and $\mathbf{A}^T$. Solving this system provides $\mathbf{u}$ and $\boldsymbol{\lambda}$. The latter can be interpreted as forces of constraint in the following sense: a removed constraint can be replaced by a system of forces characterized by $\boldsymbol{\lambda}$ multiplied by the constraint coefficients. More precisely, the constraint forces are $-\mathbf{A}^T\boldsymbol{\lambda}$.

§9.3.4. Assessment of the Lagrange Multiplier Method

In contrast to the penalty method, the method of Lagrange multipliers has the advantage of being exact (aside from computational errors due to finite precision arithmetic). It provides directly the constraint forces, which are of interest in many applications. It does not require guesses as regards weights. As the penalty method, it can be extended without difficulty to nonlinear constraints.

It is not free of disadvantages. It introduces additional unknowns, requiring expansion of the original stiffness method, and more complicated storage allocation procedures. It renders the augmented stiffness matrix indefinite, an effect that may cause grief with some linear equation solving methods that rely on positive definiteness. Finally, as the master-slave method, it is sensitive to the degree of linear independence of the constraints: if the constraint $u_2 = u_6$ is specified twice, the bordered stiffness is obviously singular.

On the whole this method appears to be the most elegant one for a general-purpose finite element program that is supposed to work as a “black box” by minimizing guesses and choices from its users. Its implementation, however, is not simple. Special care must be exercised to detect singularities due to constraint dependency and to account for the effect of loss of positive definiteness of the bordered stiffness on equation solvers.

§9.4. *The Augmented Lagrangian Method

The general matrix forms of the penalty function and Lagrangian multiplier methods are given by expressions (9.13) and (9.22), respectively. A useful connection between these methods can be established as follows.

Because the lower diagonal block of the bordered stiffness matrix in (9.22) is null, it is not possible to directly eliminate $\boldsymbol{\lambda}$. To make this possible, replace this block by $\epsilon \mathbf{S}^{-1}$, where $\mathbf{S}$ is a constraint-scaling diagonal matrix of appropriate order and ϵ is a small number. The reciprocal of ϵ is a large number called $w = 1/\epsilon$. To maintain exactness of the second equation, $\epsilon \mathbf{S}^{-1} \boldsymbol{\lambda}$ is added to the right-hand side:

$$\begin{bmatrix} \mathbf{K} & \mathbf{A}^T \\ \mathbf{A} & \epsilon \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \epsilon \mathbf{S}^{-1} \boldsymbol{\lambda}^P \end{bmatrix} \tag{9.23}$$

Here superscript P (for "predicted value") is attached to the $\boldsymbol{\lambda}$ on the right-hand side as a "tracer." We can now formally solve for $\boldsymbol{\lambda}$ and subsequently for $\mathbf{u}$. The results may be presented as

$$\begin{aligned} (\mathbf{K} + w\mathbf{A}^T\mathbf{S}\mathbf{A})\,\mathbf{u} &= \mathbf{f} + w\mathbf{A}^T\mathbf{S}\mathbf{b} - \mathbf{A}^T\boldsymbol{\lambda}^P, \\ \boldsymbol{\lambda} &= \boldsymbol{\lambda}^P + w\mathbf{S}(\mathbf{b} - \mathbf{A}\mathbf{u}), \end{aligned} \tag{9.24}$$

Setting $\boldsymbol{\lambda}^P = \mathbf{0}$ in the first matrix equation yields

$$(\mathbf{K} + w\mathbf{A}^T\mathbf{S}\mathbf{A})\,\mathbf{u} = \mathbf{f} + w\mathbf{A}^T\mathbf{S}\mathbf{b}. \tag{9.25}$$

On taking $\mathbf{W} = w\mathbf{S}$, the general matrix equation (9.13) of the penalty method is recovered.

This relation suggests the construction of *iterative procedures* in which one tries to *improve the accuracy of the penalty function method while w is kept constant* [**?**]. This strategy circumvents the aforementioned ill-conditioning problems when the weight w is gradually increased. One such method is easily constructed by inspecting (9.24). Using superscript k as an iteration index and keeping w fixed, solve equations (9.24) in tandem as follows:

$$\begin{aligned} (\mathbf{K} + \mathbf{A}^T\mathbf{W}\mathbf{A})\,\mathbf{u}^k &= \mathbf{f} + \mathbf{A}^T\mathbf{W}\mathbf{b} - \mathbf{A}^T\boldsymbol{\lambda}^k, \\ \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \mathbf{W}(\mathbf{b} - \mathbf{A}\mathbf{u}^k), \end{aligned} \tag{9.26}$$

for $k = 0, 1, \ldots,$ beginning with $\boldsymbol{\lambda}^0 = \mathbf{0}$. Then $\mathbf{u}^0$ is the penalty solution. If the process converges one recovers the exact Lagrangian solution without having to solve the Lagrangian system (9.23) directly.

The family of iterative procedures that may be precipitated from (9.24) collectively pertains to the class of *augmented Lagrangian methods*.

§9.5. Summary

The treatment of linear MFCs in finite element systems can be carried out by several methods. Three of these: master-slave elimination, penalty augmentation and Lagrange multiplier adjunction, have been discussed. It is emphasized that no method is uniformly satisfactory in terms of generality, robustness, numerical behavior and simplicity of implementation.

Figure 9.4 gives an assessment of the three techniques in terms of seven attributes.

For a general purpose program that tries to attain "black box" behavior (that is, minimal decisions on the part of users) the method of Lagrange multipliers has the edge. This edge is unfortunately blunted by a fairly complex computer implementation and by the loss of positive definiteness in the bordered stiffness matrix.

	Master-Slave Elimination	Penalty Function	Lagrange Multipliers
Generality	**fair**	**excellent**	**excellent**
Ease of implementation	**poor to fair**	**good**	**fair**
Sensitivity to user decisions	**high**	**high**	**small to none**
Accuracy	**variable**	**mediocre**	**excellent**
Sensitivity as regards constraint dependence	**high**	**none**	**high**
Retains positive definiteness	**yes**	**yes**	**no**
Modifies unknown vector	**yes**	**no**	**yes**

FIGURE 9.4. Assessment summary of three MFC application methods.

Notes and Bibliography

A form of the penalty function method, quite close to that described in §9.1.5, was first proposed by Courant in the early 1940s [154]. It entered the FEM through the work of researchers in the 1960s. There is a good description in the book by Zienkiewicz and Taylor [837].

The Lagrange Multiplier method is much older. Multipliers (called initially “coefficients”) were described by Lagrange in his famous *Mécanique Analytique* monograph [435], as part of the procedure for forming the function now called the Lagrangian. Its use in FEM is more recent than penalty methods.

Augmented Lagrangian methods have received much attention since the late 1960s, when they originated in the field of constrained optimization. The original papers are by Hestenes [360] and Powell [598]. The use of the Augmented Lagrangian Multiplier method for FEM kinematic constraints is first discussed in [212], wherein the iterative algorithm (9.26) for the master stiffness equations is derived.

References

Referenced items have been moved to Appendix R.

Homework Exercises for Chapter 9
MultiFreedom Constraints II

EXERCISE 9.1 [C+N:20] This is identical to Exercise 8.1, except that the MFC $u_2 - u_6 = 1/5$ is to be treated by the penalty function method. Take the weight w to be 10^k, in which k varies as $k = 3, 4, 5, \ldots 16$. For each sample w compute the Euclidean-norm solution error $e(w) = ||\mathbf{u}^p(w) - \mathbf{u}^{ex}||_2$, where $\mathbf{u}^p$ is the computed solution and $\mathbf{u}^{ex}$ is the exact solution listed in (E8.1). Plot $k = \log_{10} w$ versus $\log_{10} e$ and report for which weight e attains a minimum. (See Slide #5 for a check). Does it roughly agree with the square root rule (§9.1.3) if the computations carry 16 digits of precision?

As in Exercise 8.1, use *Mathematica*, *Matlab* (or similar) to do the algebra. For example, the following *Mathematica* script solves this Exercise:

```
(* Exercise 9.1 - Penalty Method  *)
(* MFC: u2-u6=1/5  variable w *)
K=MasterStiffnessOfSixElementBar[100];
Print["Stiffness K=",K//MatrixForm];
f={1,2,3,4,5,6,7}; Print["Applied forces=",f];
uexact= {0,0.27,0.275,0.25,0.185,0.07,0.14}; ew={};
For [w=100, w<=10^16, w=10*w; (* increase w by 10 every pass *)
      Khat=K; fhat=f;
      Khat[[2,2]]+=w; Khat[[6,6]]+=w; Khat[[6,2]]=Khat[[2,6]]-=w;
      fhat[[2]]+=(1/5)*w; fhat[[6]]-=(1/5)*w; (*insert penalty *)
      {Kmod,fmod}=FixLeftEndOfSixElementBar[Khat,fhat];
      u=LinearSolve[N[Kmod],N[fmod]];
      Print["Weight w=",N[w]//ScientificForm," u=",u//InputForm];
      e=Sqrt[(u-uexact).(u-uexact)];
      (*Print["L2 solution error=",e//ScientificForm]; *)
      AppendTo[ew,{Log[10,w],Log[10,e]}];
    ];
ListPlot[ew,AxesOrigin->{5,-8},Frame->True, PlotStyle->
   {AbsolutePointSize[4],AbsoluteThickness[2],RGBColor[1,0,0]},
    PlotJoined->True,AxesLabel->{"Log10(w)","Log10(u error)"}];
```

Here `MasterStiffnessOfSixElementBar` and `FixLeftEndOfSixElementBar` are the same modules listed in Exercise 8.1.

Note: If you run the above program, you may get several beeps from *Mathematica* as it is processing some of the systems with very large weights. Don't be alarmed: those are only warnings. The `LinearSolve` function is alerting you that the coefficient matrices $\hat{\mathbf{K}}$ for weights of order 10^{12} or bigger are ill-conditioned.

EXERCISE 9.2 [C+N:15] Again identical to Exercise 8.1, except that the MFC $u_2 - u_6 = 1/5$ is to be treated by the Lagrange multiplier method. The results for the computed $\mathbf{u}$ and the recovered force vector $\mathbf{Ku}$ should agree with (E8.1). Use *Mathematica*, *Matlab* (or similar) to do the algebra. For example, the following *Mathematica* script solves this Exercise:

```
(* Exercise 9.2 - Lagrange Multiplier Method  *)
(* MFC: u2-u6=1/5  *)
K=MasterStiffnessOfSixElementBar[100];
Khat=Table[0,{8},{8}]; f={1,2,3,4,5,6,7}; fhat=AppendTo[f,0];
For [i=1,i<=7,i++, For[j=1,j<=7,j++, Khat[[i,j]]=K[[i,j]] ]];
{Kmod,fmod}=FixLeftEndOfSixElementBar[Khat,fhat];
```

```
Kmod[[2,8]]=Kmod[[8,2]]= 1;
Kmod[[6,8]]=Kmod[[8,6]]=-1; fmod[[8]]=1/5;
Print["Kmod=",Kmod//MatrixForm];
Print["fmod=",fmod];
umod=LinearSolve[N[Kmod],N[fmod]]; u=Take[umod,7];
Print["Solution u=",u ,",   lambda=",umod[[8]]];
Print["Recovered node forces=",K.u];
```

Here `MasterStiffnessOfSixElementBar` and `FixLeftEndOfSixElementBar` are the same modules listed in Exercise 8.1.

Does the computed solution agree with (E8.1)?

EXERCISE 9.3 [A:10] For the example structure, show which penalty elements would implement the following MFCs:

$$\begin{aligned} &\text{(a)} \quad u_2 + u_6 = 0, \\ &\text{(b)} \quad u_2 - 3u_6 = 1/3. \end{aligned} \tag{E9.1}$$

As answer, show the stiffness equations of those two elements in a manner similar to (9.1).

EXERCISE 9.4 [A/C+N:15+15+10] Suppose that the assembled stiffness equations for a one-dimensional finite element model before imposing constraints are

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}. \tag{E9.2}$$

This system is to be solved subject to the multipoint constraint

$$u_1 = u_3. \tag{E9.3}$$

(a) Impose the constraint (E9.3) by the master-slave method taking u_1 as master, and solve the resulting 2×2 system of equations by hand.

(b) Impose the constraint (E9.3) by the penalty function method, leaving the weight w as a free parameter. Solve the equations by hand or CAS (Cramer's rule is recommended) and verify analytically that as $w \to \infty$ the solution approaches that found in (a). Tabulate the values of u_1, u_2, u_3 for $w = 0,\ 1,\ 10,\ 100$. *Hint 1*: the value of u_2 should not change. *Hint 2*: the solution for u_1 should be $(6w + 5)/(4w + 4)$.

(c) Impose the constraint (E9.3) by the Lagrange multiplier method. Show the 4×4 multiplier-augmented system of equations analogous to (9.13) and solve it by computer or calculator.

EXERCISE 9.5 [A/C:10+15+10] The left end of the cantilevered beam-column member illustrated in Figure E9.1 rests on a skew-roller that forms a 45° angle with the horizontal axis x. The member is loaded axially by a force P as shown. The finite element equations upon removing the fixed right end freedoms $\{u_{x2}, u_{y2}, \theta_2\}$, but *before* imposing the skew-roller MFC, are

$$\begin{bmatrix} EA/L & 0 & 0 \\ 0 & 12EI/L^3 & 6EI/L^2 \\ 0 & 6EI/L^2 & 4EI/L \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ \theta_1 \end{bmatrix} = \begin{bmatrix} P \\ 0 \\ 0 \end{bmatrix}, \tag{E9.4}$$

where E, A, and $I = I_{zz}$ are given member properties, θ_1 is the left end rotation, and L is the member length.[7]

[7] The stiffness equations for a beam column are derived in Part III of this book. For now consider (E9.4) as a recipe.

To simplify the calculations set $P = \alpha EA$, and $I = \beta AL^2$, in which α and β are dimensionless parameters, and express the following solutions in terms of α and β.

(a) Apply the skew-roller constraint by the master-slave method (make u_{y1} slave) and solve for u_{x1} and θ_1 in terms of L, α and β. This may be done by hand or a CAS. Partial solution: $u_{x1} = \alpha L/(1+3\beta)$.

(b) Apply the skew-roller constraint with the penalty method by inserting a penalty element at node 1. Follow the rule of §9.1.4 to construct the 2×2 penalty stiffness. Compute u_{x1} from the modified equations (Cramer's rule is recommended if solved by hand). Verify that as $w \to \infty$ the answer obtained in (a) is recovered. Partial solution: $u_{x1} = \alpha L(3EA\beta + wL)/(3EA\beta + wL(1+3\beta))$. Can the penalty stiffness be physically interpreted in some way?

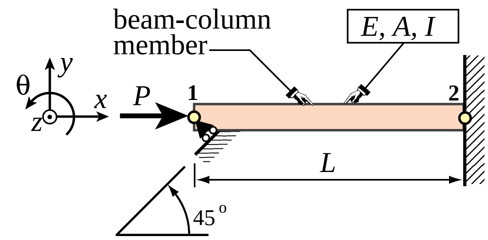

FIGURE E9.1. Cantilevered beam-column on skew-roller for Exercise 9.5.

(c) Apply the skew roller constraint by Lagrangian multiplier adjunction, and solve the resulting 4×4 system of equations using a CAS (by hand it will take long). Verify that you get the same solution as in (a).

EXERCISE 9.6 [A:5+5+10+10+5] A cantiveler beam-column is to be joined to a plane stress plate mesh as depicted in Figure E9.2.[8] Both pieces move in the plane $\{x, y\}$. Plane stress elements have two degrees of freedom per node: two translations u_x and u_y along x and y, respectively, whereas a beam-column element has three: two translations u_x and u_y along x and y, and one rotation (positive CCW) θ_z about z. To connect the cantilever beam to the mesh, the following "gluing" conditions are applied:

(1) The horizontal (u_x) and vertical (u_y) displacements of the beam at their common node (2 of beam, 4 of plate) are the same.

(2) The beam end rotation θ_2 and the mean rotation of the plate edge 3–5 are the same. For infinitesimal displacements and rotations the latter is $\theta_{35}^{avg} = (u_{x5} - u_{x3})/H$.

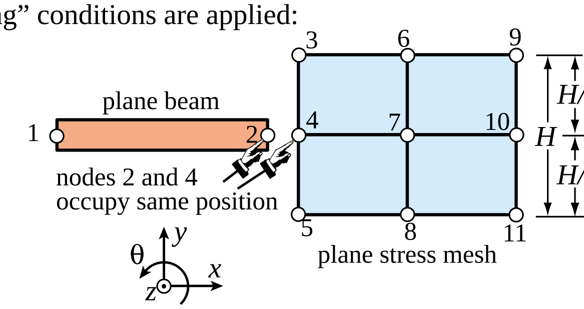

FIGURE E9.2. Beam linked to plate in plane stress for Exercise 9.6. Beam shown slightly separate from plate for visualization convenience: nodes 2 and 4 actually are at the same location.

Questions:

(a) Write down the three MFC conditions: two from (1) and one from (2), and state whether they are linear and homogeneous.

(b) Where does the above expression of θ_{35}^{avg} come from? (Geometric interpretation is OK.) Can it be made more accurate[9] by including u_{x4}?

(c) Write down the master-slave transformation matrix if $\{u_{x2}, u_{y2}, \theta_2\}$ are picked as slaves. It is sufficient to write down the transformation for the DOFs of nodes 2, 3, 4, and 5, which gives a $\mathbf{T}$ of order 9×6, since the transformations for the other freedoms are trivial.

(d) If the penalty method is used, write down the stiffness equations of the three penalty elements assuming the same weight w is used. Their stiffness matrices are of order 2×2, 2×2 and 3×3, respectively. (Do not proceed further)

[8] This is extracted from a question previously given in the Aerospace Ph. D. Preliminary Exam. Technically it is not difficult once the student understand what is being asked. This can take some time, but a HW is more relaxed.

[9] To answer the second question, observe that the displacements along 3–4 and 4–5 vary linearly. Thus the angle of rotation about z is constant for each of them, and (for infinitesimal displacements) may be set equal to the tangent.

(e) If Lagrange multiplier adjunction is used, how many Lagrange multipliers will you need to append? (Do not proceed further).

EXERCISE 9.7 [A:30] Show that the master-slave transformation method $\mathbf{u} = \mathbf{T}\hat{\mathbf{u}}$ can be written down as a special form of the method of Lagrange multipliers. Start from the augmented functional

$$\Pi_{MS} = \tfrac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} + \boldsymbol{\lambda}^T(\mathbf{u} - \mathbf{T}\hat{\mathbf{u}}) \tag{E9.5}$$

and write down the stationarity conditions of Π_{MS} with respect to $\mathbf{u}$, $\boldsymbol{\lambda}$ and $\hat{\mathbf{u}}$ in matrix form.

EXERCISE 9.8 [A:35] Check the matrix equations (9.23) through (9.26) quoted for the Augmented Lagrangian method.

EXERCISE 9.9 [A:40] (Advanced, close to a research paper). Show that the master-slave transformation method $\mathbf{u} = \mathbf{T}\hat{\mathbf{u}}$ can be expressed as a limit of the penalty function method as the weights go to infinity. Start from the augmented functional

$$\Pi_P = \tfrac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} + \tfrac{1}{2}w(\mathbf{u} - \mathbf{T}\hat{\mathbf{u}})^T(\mathbf{u} - \mathbf{T}\hat{\mathbf{u}}) \tag{E9.6}$$

Write down the matrix stationarity conditions with respect to to $\mathbf{u}$ and $\hat{\mathbf{u}}$ and take the limit $w \to \infty$. *Hint*: using Woodbury's formula (Appendix C, §C.5.2)

$$(\mathbf{K} + w\mathbf{T}^T\mathbf{S}\mathbf{T})^{-1} = \mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{T}^T(\overline{\mathbf{K}} + w^{-1}\mathbf{S}^{-1})^{-1}\mathbf{T}\,\mathbf{K}^{-1}. \tag{E9.7}$$

show that

$$\overline{\mathbf{K}}^{-1} = \mathbf{T}\mathbf{K}^{-1}\mathbf{T}^T. \tag{E9.8}$$