Lecture #26

Introduction to Variational Methods

Outline

• Overview
• Galerkin Method
• Example
• Finite Element Method
• Method of Moments
• Other Worthy Methods
  – Boundary element method
  – Spectral domain method
## Overview

<table>
<thead>
<tr>
<th>Method</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finite Element Method</strong></td>
<td>Utilizes a volume mesh, matrices are sparse, requires boundary conditions</td>
</tr>
<tr>
<td><strong>Boundary Element Method</strong></td>
<td>Utilizes a conformal surface mesh, matrices are full, good for surface devices, does not require boundary conditions</td>
</tr>
<tr>
<td><strong>Method of Moments</strong></td>
<td>Typically makes a PEC approximation, utilizes a conformal surface mesh, can be 1D for thin wire structures, matrices are full, good for surface devices, does not require boundary conditions</td>
</tr>
<tr>
<td><strong>Spectral Domain Method</strong></td>
<td>BEM/MoM in Fourier-Space, matrices are full, excellent for periodic structures, does not require boundary conditions</td>
</tr>
</tbody>
</table>

[Image of the slide]
Both the variational method and the method of weighted residuals can be used to write a governing equation in matrix form.

Both approaches yield exactly the same matrices.

The Galerkin method is the most popular special case of weighted residual methods.

Galerkin Method

Boris Grigoryevich Galerkin
1871 - 1945
Linear Equations

Consider the following linear homogeneous equation.

\[
L \left[ f(x) \right] = g(x)
\]

- \( L[\ ] \) = Linear operation
- \( f(x) \) = unknown solution
- \( g(x) \) = known driving function

The linear operator \( L[\] \) has the following properties:

\[
L \left[ f_1(x) + f_2(x) \right] = L \left[ f_1(x) \right] + L \left[ f_2(x) \right]
\]
\[
L \left[ a f(x) \right] = a L \left[ f(x) \right]
\]
\[
L_1 \left[ L_2 \left[ f(x) \right] \right] = L_2 \left[ L_1 \left[ f(x) \right] \right]
\]

Inner Product

An inner product is a scalar quantity that provides a measure of similarity between two functions.

\[
\langle f(x), g(x) \rangle \equiv \text{inner product between } f(x) \text{ and } g(x)
\]

- \( \langle f, g \rangle = 0 \quad \text{if } f \text{ and } g \text{ are orthogonal} \)
- \( \langle f, g \rangle = \text{small number} \quad \text{if } f \text{ and } g \text{ are very different} \)
- \( \langle f, g \rangle = \text{big number} \quad \text{if } f \text{ and } g \text{ are very similar} \)

It is common to scale functions such that \( \langle f, f \rangle = 1 \).

An appropriate inner product must satisfy

\[
\langle f, g \rangle = \langle g, f \rangle
\]
\[
\langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle
\]
\[
\langle f, f^* \rangle \begin{cases} > 0 & f \neq 0 \\ = 0 & f = 0 \end{cases}
\]

We will use the following inner product:

\[
\langle f(z), g(z) \rangle = \int f(z) g(z) dz
\]
Expansion Into a Set of Basis Functions

Let \( f(x) \) be expanded into a set of basis functions.

\[
f(x) = \sum_{n} a_n v_n(x)
\]

- \( f(x) \) = unknown function
- \( a_n \) = coefficient of \( n^{th} \) basis function
- \( v_n(x) \) = \( n^{th} \) basis function

We choose the basis functions with two considerations: (1) ease of calculations, and (2) minimize how many are needed in the expansion to accurately portray the field.

Linear Equation in Terms of Basis Functions

First we substitute the expansion into the original linear equation.

\[
L[f(x)] = g(x)
\]

Using the properties of linear operations, we get

\[
L[\sum_{n} a_n v_n(x)] = g(x)
\]

\[
\sum_{n} L[a_n v_n(x)] = g(x)
\]

We try to choose \( v_n(x) \) so that \( L[v_n(x)] \) is easy to calculate and efficiently represents \( f(x) \).
Method of Weighted Residuals (1 of 2)

Similar to how we choose a set of basis functions, we choose another set of weighting functions.

\[ w_n(x) \]

We start with our linear inhomogeneous equation and calculate the inner product with \( w_m(x) \) of both sides. Here we “test” both sides with \( w_m(x) \).

\[
\sum_n a_n L[v_n(x)] = g(x) \\
\left< w_m(x), \sum_n a_n L[v_n(x)] \right> = \left< w_m(x), g(x) \right> \\
\sum_n a_n \left< w_m(x), L[v_n(x)] \right> = \left< w_m(x), g(x) \right>
\]

Method of Weighted Residuals (2 of 2)

This equation can be written in matrix form as

\[
\sum_n a_n \left< w_m(x), L[v_n(x)] \right> = \left< w_m(x), g(x) \right> \rightarrow \begin{bmatrix} z_{m1} & \cdots & z_{mM} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \langle w_1, g \rangle \\ \langle w_2, g \rangle \\ \vdots \\ \langle w_M, g \rangle \end{bmatrix}
\]

\[
[a_n] = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \quad [g_m] = \begin{bmatrix} \langle w_1, g \rangle \\ \langle w_2, g \rangle \\ \vdots \\ \langle w_M, g \rangle \end{bmatrix}
\]
Galerkin Method

The Galerkin method is the method of weighted residuals, but the weighting functions are made to be the same as the basis functions.

\[ w_m(x) = v_m(x) \]

The matrix equation becomes

\[
\sum_n a_n \langle v_m(x), L[v_n(x)] \rangle = \langle v_m(x), g(x) \rangle \rightarrow \begin{bmatrix} z_{mn} \end{bmatrix} \begin{bmatrix} a_n \end{bmatrix} = \begin{bmatrix} g_m \end{bmatrix}
\]

Summary of the Galerkin Method

The Galerkin method can be used to find the solution to any linear inhomogeneous equation.

\[ L[f] = g \]

Step 1 – Expand the unknown function into a set of basis functions.

\[ f = \sum_n a_n v_n \quad L[\sum_n a_n v_n] = g \rightarrow \sum_n a_n L[v_n] = g \]

Step 2 – Test both sides of the equation with the basis functions using an inner product

\[ \langle v_n, \sum_n a_n L[v_n] \rangle = \langle v_n, g \rangle \rightarrow \sum_n a_n \langle v_n, L[v_n] \rangle = \langle v_n, g \rangle \]

Step 3 – Form a matrix equation

\[
\begin{bmatrix} z_{mn} \end{bmatrix} \begin{bmatrix} a_n \end{bmatrix} = \begin{bmatrix} g_m \end{bmatrix}
\]
Example

Governing Equation and Its Solution

**Governing Equation**

We will apply the Galerkin method to solve the following differential equation.

\[-\frac{d^2 f}{dx^2} = 1 + 4x^2\]

\[f(0) = f(1) = 0\quad 0 \leq x \leq 1\]

**Simple Analytical Solution**

This is a simple boundary value problem with the following solution.

\[f(x) = \frac{5x}{6} - \frac{x^2}{2} - \frac{x^4}{3}\]

We wish to solve this using the Galerkin method.
Choose Basis Functions

Basis Functions
For this problem, it will be convenient to choose as basis functions:

\[ v_n = x - x^{n+1} \]

Expansion of the Function into the Basis
We expand our function into this set of basis functions as

\[ f(x) = \sum_{n=1}^{N} a_n \left( x - x^{n+1} \right) \]

Choose Testing Functions

Testing Functions
The Galerkin method uses the same function for basis and testing.

\[ w_n = v_n = x - x^{n+1} \]
Form of Final Solution

Recall that we are converting a linear equation into a matrix equation according to

\[ L[f] = g \quad \implies \quad [z_{mn}][a_n] = [g_m] \]

\[ [z_{mn}] = \begin{bmatrix} \langle v_1, L[v_1] \rangle & \langle v_1, L[v_2] \rangle \\ \langle v_2, L[v_1] \rangle & \langle v_2, L[v_2] \rangle \\ \vdots & \vdots \end{bmatrix} \]

\[ [a_n] = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad \text{and} \quad [g_m] = \begin{bmatrix} \langle v_1, g \rangle \\ \langle v_2, g \rangle \\ \vdots \\ \langle v_n, g \rangle \end{bmatrix} \]

To do this, we need to evaluate \( \langle v_n, L[v_n] \rangle \) and \( \langle v_n, g \rangle \).

First Inner Product

\[ \langle v_n, L[v_n] \rangle = \int v_n L[v_n] \, dx \]

\[ = \int \left( x^n \right) \left( \frac{d^2}{dx^2} \left( x^n \right) \right) \, dx \]

\[ = -\int \left( x^n \right) \left( -n(n+1)x^{n-1} \right) \, dx \]

\[ = n(n+1) \int \left( x^n - x^{n-1} \right) \, dx \]

\[ = n(n+1) \left[ \frac{x^{n+1}}{n+1} \frac{x^{n+1}}{m+n+1} \right] \]

\[ = n(n+1) \left[ \frac{1}{n+1} \frac{1}{m+n+1} \right] \]

\[ = n(n+1) \left[ \frac{m+n+1}{(n+1)(m+n+1)} \right] \]

\[ = \frac{mn}{(m+n+1)} \]

\[ z_{mn} = \langle v_n, L[v_n] \rangle = \frac{mn}{(m+n+1)} \]
Second Inner Product

\[
\langle v, g \rangle = \int v \cdot g \, dx
\]

\[
= \int (x-x^2)(1+4x^2) \, dx
\]

\[
= \int (x+4x^4-x^3-4x^3) \, dx
\]

\[
= \left[ \frac{x^2}{2} + x^4 - \frac{x^4}{m+2} - \frac{4x^4}{m+4} \right]
\]

\[
= \frac{1}{2} + 1 - \frac{1}{m+2} - \frac{4}{m+4}.
\]

\[
g_m = \langle v_m, g \rangle = \frac{m(3m+8)}{2(m+2)(m+4)}
\]

Try \( N=1 \)

For \( N=1 \), our matrix equation isn’t even a matrix equation.

\[
[z_{11}] [a] = [g_1] \quad \rightarrow \quad z_{11} a_1 = g_1
\]

Applying our equations for \( z_{mn} \) and \( g_m \), we get

\[
z_{11} = \frac{1 \cdot 1}{(1+1+1)} = \frac{1}{3}.
\]

\[
g_1 = \frac{1(3 \cdot 1 + 8)}{2(1+2)(1+4)} = \frac{11}{30}.
\]

The coefficient is

\[
a_1 = \frac{g_1}{z_{11}} = \frac{11/30}{1/3} = \frac{11}{10}.
\]

Finally, the solution for \( N=1 \) is

\[
f(x) = a_1 (x-x^2) = \frac{11x}{10} - \frac{11x^2}{10}.
\]

\[\text{Not correct. Need larger } N.\]
Try \( N=2 \)

For \( N=2 \), our matrix equation is

\[
\begin{bmatrix}
  z_{11} & z_{12} \\
  z_{21} & z_{22}
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2
\end{bmatrix}
= \begin{bmatrix}
  g_1 \\
  g_2
\end{bmatrix}
\]

Applying our equations for \( z_{mn} \) and \( g_m \), we get

\[
\begin{align*}
  z_{11} &= \frac{1}{1+1+1} = \frac{1}{3} \\
  z_{12} &= \frac{1}{1+2+1} = \frac{1}{2} \\
  z_{21} &= \frac{2}{1+2+1} = \frac{1}{2} \\
  z_{22} &= \frac{2}{1+2+1} = \frac{2}{3} \\
  z_{31} &= \frac{2}{1+2+1} = \frac{2}{3} \\
  z_{32} &= \frac{2}{1+2+1} = \frac{4}{3}
\end{align*}
\]

\[
\begin{align*}
  g_1 &= \frac{(3+1+8)}{2(1+2)(1+4)} = \frac{11}{30} \\
  g_2 &= \frac{2(3+2+8)}{2(1+2)(1+4)} = \frac{23}{12}
\end{align*}
\]

The coefficients are

\[
\begin{bmatrix}
  a_1 \\
  a_2
\end{bmatrix}
= \begin{bmatrix}
  z_{11} & z_{12} \\
  z_{21} & z_{22}
\end{bmatrix}^{-1}
\begin{bmatrix}
  g_1 \\
  g_2
\end{bmatrix}
= \begin{bmatrix}
  \frac{1}{3} & \frac{2}{3} \\
  \frac{1}{2} & \frac{1}{2}
\end{bmatrix}^{-1}
\begin{bmatrix}
  \frac{1}{3} & \frac{1}{2} & \frac{1}{3} \\
  \frac{2}{3} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
= \begin{bmatrix}
  \frac{1}{3} & \frac{4}{9} & \frac{1}{9} \\
  \frac{2}{3} & \frac{1}{9} & \frac{4}{9}
\end{bmatrix}
\]

Finally, the solution for \( N=2 \) is

\[
f(x) = a_1(x - x^3) + a_2(x - x^3) = \frac{23x}{30} - \frac{x^2}{10} - \frac{2x^3}{3}
\]

Still not correct. Need larger \( N \).

Try \( N=3 \)

For \( N=3 \), our matrix equation is

\[
\begin{bmatrix}
  z_{11} & z_{12} & z_{13} \\
  z_{21} & z_{22} & z_{23} \\
  z_{31} & z_{32} & z_{33}
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  a_3
\end{bmatrix}
= \begin{bmatrix}
  g_1 \\
  g_2 \\
  g_3
\end{bmatrix}
\]

Applying our equations for \( z_{mn} \) and \( g_m \), we get

\[
\begin{align*}
  \begin{bmatrix}
  \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \\
  \frac{1}{2} & \frac{1}{3} & \frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  a_3
\end{bmatrix}
&= \begin{bmatrix}
  \frac{41}{30} \\
  \frac{2}{12} \\
  \frac{51}{70}
\end{bmatrix} \\
\Rightarrow \\
\begin{bmatrix}
  \frac{1}{3} & \frac{1}{2} & \frac{1}{3} \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \\
  \frac{1}{2} & \frac{1}{3} & \frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
  a_1 \\
  a_2 \\
  a_3
\end{bmatrix}
&= \begin{bmatrix}
  \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\
  \frac{1}{2} & \frac{1}{2} & 1 \\
  \frac{1}{2} & \frac{1}{3} & \frac{1}{2}
\end{bmatrix}^{-1}
\begin{bmatrix}
  \frac{41}{30} \\
  \frac{2}{12} \\
  \frac{51}{70}
\end{bmatrix}
= \begin{bmatrix}
  \frac{1}{3} \\
  \frac{2}{3} \\
  \frac{1}{2}
\end{bmatrix}
\]

Finally, the solution for \( N=3 \) is

\[
f(x) = a_1(x - x^3) + a_2(x - x^3) + a_3(x - x^3) = \frac{5x}{6} - \frac{x^2}{2} - \frac{x^4}{3}
\]

Exact solution!
Try $N=4$

For $N=4$, we get a larger matrix equation, but it also converges to the exact solution.

In fact, the method converges to the exact solution for all $N \geq 3$.

\[ f(x) = \sum_{n=1}^{N} a_n (x - x^n) = \frac{5x}{6} - \frac{x^2}{2} - \frac{x^4}{3} \]

Exact solution

Finite Element Method
What is a Finite Element?

Like the finite-difference method, the finite element method (FEM) discretizes the problem space. In FEM, it is divided into small regions called finite elements.

Meshing

Meshing describes the “intelligent” process of subdividing space into non-overlapping finite elements. Usually this is done to conform perfectly to the shapes of devices.

- FEM mesh conforms very well to curved geometries.
- Mesh can be locally refined to resolve small features or abruptly varying fields.
- Adaptive meshing – Refines the mesh based on a prior solution.
Adaptive Meshing

A model is performed using a coarse mesh. The mesh is then refined where the field is varying rapidly and the model is run again. This continues until convergence.

FEM Vs. Finite-Difference Grids

The finite element mesh offers minimal discretization error.

Node Elements Vs. Edge Elements

Node Elements

• Field is known and stored at the element nodes.
• Elsewhere, field is interpolated.
• Suffers from spurious solutions due to lack of enforcing divergence conditions.
• Matrices better conditioned.

Edge Elements

• Vector field is known at the edges of the elements.
• Elsewhere, field is interpolated.
• Solves the spurious solution problem because the divergence conditions are enforced through continuity of the tangential field components.
• Matrices have poorer conditioning.

Shape Functions

The field within the elements is expanded into a set of basis functions that are called “shape” functions.

\[ E(x, y) = \sum_{i=1}^{3} u_i N_i(x, y) \]

For linear triangular elements, the shape functions are

\[ N_i(x, y) = a_i x + b_i y + c_i \]


**Element Matrix K**

The element matrix is \( N \times N \) where \( N \) is the number of nodes in an element. It is derived to enforce Maxwell’s equations within an element.

\[
\nabla \times \left( \frac{1}{\mu} \nabla \times \vec{E} \right) - \omega^2 \epsilon \vec{E} = 0
\]

Galeraik Method  
Variational Method

\[
\begin{bmatrix}
K_{11}^{(e)} & K_{12}^{(e)} & K_{13}^{(e)} \\
K_{21}^{(e)} & K_{22}^{(e)} & K_{23}^{(e)} \\
K_{31}^{(e)} & K_{32}^{(e)} & K_{33}^{(e)}
\end{bmatrix}
\begin{bmatrix}
u_1^{(e)} \\
u_2^{(e)} \\
u_3^{(e)}
\end{bmatrix}
= \begin{bmatrix}
b_1^{(e)} \\
b_2^{(e)} \\
b_3^{(e)}
\end{bmatrix}
\]

Nodes are normally labeled going counter-clockwise.

---

**Global Matrix**

The global matrix is the overall matrix wave equation.

\[
\nabla \times \left( \mu^{-1} \nabla \times \vec{E} \right) - \omega^2 \epsilon \vec{E} = \vec{b}
\]

\[
[K] \begin{bmatrix} \vec{E} \end{bmatrix} = \begin{bmatrix} \vec{b} \end{bmatrix}
\]

\[
[K] = \begin{bmatrix}
\end{bmatrix}
\]

\[
[b] = \begin{bmatrix}
? \\
? \\
? \\
? \\
? \\
? \\
? \\
? \\
\end{bmatrix}
\]
Assembling the Global Matrix (1 of 5)

The global matrix is assembled by adding the elements of the element matrices to the corresponding elements of the global matrix.

\[
[K] = \sum_{e=1}^{N} [K^{(e)}] \quad [K] = 8 \times 8 \text{ global matrix} \\
[K]^{(e)} = 4 \times 4 \text{ element matrix}
\]

The global matrix is initialized as

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}
\]

Assembling the Global Matrix (2 of 5)

Add Element 1
1. We match the local nodes to the global nodes.

<table>
<thead>
<tr>
<th>Local Node Number</th>
<th>Global Node Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

\[
K^{(e)}_{ij} \rightarrow K^{(\text{global})}_{mn} \\
i \rightarrow m \\
j \rightarrow n
\]

We use this chart to map local matrix elements to global matrix elements.
Assembling the Global Matrix (3 of 5)

Add Element 1

2. We add \([K^{(1)}]\) to the global matrix.

\[
K = \begin{bmatrix}
K^{(1)}_{11} & K^{(1)}_{12} & K^{(1)}_{13} & 0 & 0 & 0 & K^{(1)}_{17} \\
K^{(1)}_{21} & K^{(1)}_{22} & K^{(1)}_{23} & 0 & 0 & 0 & K^{(1)}_{27} \\
K^{(1)}_{31} & K^{(1)}_{32} & K^{(1)}_{33} & 0 & 0 & 0 & K^{(1)}_{37} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K^{(1)}_{71} & K^{(1)}_{72} & K^{(1)}_{73} & 0 & 0 & 0 & K^{(1)}_{77}
\end{bmatrix}
\]

Assembling the Global Matrix (4 of 5)

Add Element 2

1. Match local to global node numbers.
2. Add \([K^{(2)}]\) to the global matrix.

\[
K = \begin{bmatrix}
K^{(2)}_{11} & K^{(2)}_{12} & K^{(2)}_{13} & 0 & 0 & 0 & K^{(2)}_{17} \\
K^{(2)}_{21} & K^{(2)}_{22} & K^{(2)}_{23} & 0 & 0 & 0 & K^{(2)}_{27} \\
K^{(2)}_{31} & K^{(2)}_{32} & K^{(2)}_{33} + K^{(2)}_{14} & K^{(2)}_{34} & K^{(2)}_{35} & K^{(2)}_{36} & K^{(2)}_{37} \\
0 & 0 & K^{(2)}_{14} & K^{(2)}_{15} & K^{(2)}_{16} & K^{(2)}_{17} & 0 \\
0 & 0 & K^{(2)}_{24} & K^{(2)}_{25} & K^{(2)}_{26} & K^{(2)}_{27} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K^{(2)}_{71} & K^{(2)}_{72} & K^{(2)}_{73} + K^{(2)}_{14} & K^{(2)}_{74} & K^{(2)}_{75} & K^{(2)}_{76} & 0 & K^{(2)}_{77}
\end{bmatrix}
\]
Assembling the Global Matrix (5 of 5)

Add Element 3
1. Match local to global node numbers.
2. Add \([K^{(3)}]\) to the global matrix.

\[
K = \begin{bmatrix}
K_{11}^{(3)} & K_{12}^{(3)} & K_{13}^{(3)} & 0 & 0 & 0 & K_{17}^{(3)} \\
K_{21}^{(3)} & K_{22}^{(3)} & K_{23}^{(3)} & 0 & 0 & 0 & K_{27}^{(3)} \\
K_{31}^{(3)} & K_{32}^{(3)} & K_{33}^{(3)} & 0 & 0 & 0 & K_{37}^{(3)} \\
0 & 0 & 0 & K_{44}^{(3)} & K_{45}^{(3)} & K_{46}^{(3)} & K_{47}^{(3)} \\
0 & 0 & 0 & K_{54}^{(3)} & K_{55}^{(3)} & K_{56}^{(3)} & K_{57}^{(3)} \\
0 & 0 & 0 & K_{64}^{(3)} & K_{65}^{(3)} & K_{66}^{(3)} & K_{67}^{(3)} \\
0 & 0 & 0 & K_{74}^{(3)} & K_{75}^{(3)} & K_{76}^{(3)} & K_{77}^{(3)} \\
K_{71}^{(3)} & K_{72}^{(3)} & K_{73}^{(3)} & K_{74}^{(3)} & K_{75}^{(3)} & K_{76}^{(3)} & K_{77}^{(3)}
\end{bmatrix}
\]

Don’t Forget about \([b]\)

We follow the same procedure for \([b]\) and get

\[
b = \begin{bmatrix}
b_1^{(1)} \\
b_4^{(1)} \\
b_3^{(1)} + b_4^{(2)} \\
b_3^{(2)} \\
b_2^{(2)} + b_3^{(3)} \\
b_2^{(3)} \\
b_1^{(3)} \\
b_2^{(1)} + b_1^{(2)} + b_4^{(3)}
\end{bmatrix}
\]
Overall Solution

We solve this problem as

\[
\begin{bmatrix} K \end{bmatrix} \begin{bmatrix} \tilde{E} \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}
\]

\[
\downarrow
\]

\[
\begin{bmatrix} \tilde{E} \end{bmatrix} = \begin{bmatrix} K \end{bmatrix}^{-1} \begin{bmatrix} b \end{bmatrix}
\]

Domain Decomposition Methods

The grid is split into a number of smaller grids. The overall solution is found by solving the smaller subdomains and iterating their solutions.
Method of Moments

Two Common Forms

The method of moments is most commonly applied to one of two special cases:

- **Thin Wire Devices**
- **Thin Metal Sheets**
Thin Wire Devices

Thin wire devices are most commonly modeled using 1D integral equations solved along the length of the wire.

Pocklington’s Integral Equation

Pocklington's integral equation is the most famous, but converges slower due to the derivative operator operating directly on a singularity in the kernel. It is the simplest to solve.

\[
E_z^{inc} = \frac{j}{\omega \epsilon} \int_{L} l_1(z') \left( k^2 + \frac{\partial^2}{\partial z'^2} \right) e^{-jkr} \frac{dz'}{4\pi r}
\]

Hallen’s Integral Equation

Hallen’s integral equations is a modified version of Pocklington’s integral equal which fixes the singularity. It is more complicated to solve, but converges faster.

\[
E_z^{inc} = \frac{j}{\omega \epsilon} \left( k^2 + \frac{\partial^2}{\partial z'^2} \right) \int_{L} l_1(z') e^{-jkr} \frac{dz'}{4\pi r}
\]

Thin Metallic Sheets

Metal sheets are most commonly modeled using 2D integral equations solved along the length of the wire.

Triangular meshes are typically used where the Rao-Wilton-Glisson (RWG) edge element is most famous.

A drawback of the method of moments is that it leads to full matrices where each element is coupled to every other element. This leads to prohibitively large matrices for very large structures.

The fast multiple method is a way of treating far away elements as a single element, thereby significantly reducing the number of elements in the final matrix.

Other Worthy Methods
Boundary Element Method

The wave equation describes the field through volumetric space. The solution requires the field throughout the volume to be resolved. This leads to a large number of elements. Matrices are sparse. Boundary conditions are needed.

Maxwell’s equations can be manipulated into the form of a surface integral. For many structures, this formulation requires many fewer elements and provides faster solution. Matrices are full. Boundary conditions are needed.

FEM Mesh
183 nodes
321 elements
sparse matrix

BEM Mesh
24 nodes
23 elements
full matrix

Spectral Domain Method

The spectral domain method is essentially the method of moments in Fourier-space. Many variations of this method exist.

Transmission Line Analysis

Array Antennas and Frequency Selective Surfaces