Lecture 36

Computational Electromagnetics, Numerical Methods

Due to the advent of digital computers and the blinding speed at which computations can be done, numerical methods to seek solutions of Maxwell's equations have become vastly popular. Massively parallel digital computers now can compute at tera\peta\exa-flops throughputs [209], where FLOPS stands for "floating operations per second". They have also spawn terms that we have not previously heard of (see also Figure 36.1).

Computer performance

Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}

Figure 36.1: Nomenclature for measuring the speed of modern day computers (courtesy of Wikipedia [209]).

We repeat a quote from Freeman Dyson—"Technology is a gift of God. After the gift of life it is perhaps the greatest of God's gifts. It is the mother of civilizations, of arts and of sciences." The spurr for computer advancement is due to the second world war. During then, men went to war while women stayed back to work as computers, doing laborious numerical computations manually (see Figure 36.2 [210]), The need for a faster computer is obvious. Unfortunately, in the last half century or so, we have been using a large part of the gift of

technology to destroy God's greatest gift, life, in warfare!

Figure 36.2: A woman working as a computer shortly after the second world war (courtesy of Wikipedia [210]).

36.1 Computational Electromagnetics and Numerical Methods

Due to the high fidelity of Maxwell's equations in describing electromagnetic physics in nature, often time, a numerical solution obtained by solving Maxwell's equations is more reliable than laboratory experiments. This field is also known as ***computational electromagnetics***. Numerical methods exploit the blinding speed of modern digital computers to perform calculations, and hence to solve large system of equations.

Computational electromagnetics consists mainly of two classes of numerical solvers: one that solves differential equations directly, the differential-equation solvers; and one that solves integral equations, the integral equation solvers. Both these classes of equations are derived from Maxwell's equations.

36.1.1 Examples of Differential Equations

An example of differential equations written in terms of sources are the scalar wave equation:

$$(\nabla^2 + k^2)\,\phi(\mathbf{r}) = Q(\mathbf{r}), \tag{36.1.1}$$

An example of vector differential equation for vector electromagnetic field is

$$\nabla \times \overline{\boldsymbol{\mu}}^{\,-1} \cdot \nabla \times \mathbf{E}(\mathbf{r}) - \omega^2 \overline{\boldsymbol{\epsilon}} \cdot \mathbf{E}(\mathbf{r}) = i\omega \mathbf{J}(\mathbf{r}) \tag{36.1.2}$$

These equations are linear equations. They have one commonality, i.e., they can be abstractly written as

$$\mathscr{L} f = g \tag{36.1.3}$$

where $\mathscr{L}$ is the differential operator which is linear, and f is the unknown, and g is the driving source. Differential equations, or partial differential equations, as mentioned before, have to be solved with boundary conditions. Otherwise, there is no unique solution to these equations.

In the case of the scalar wave equation (36.1.1), $\mathscr{L} = (\nabla^2 + k^2)$ is a differential operator. In the case of the electromagnetic vector wave equation (36.1.2), $\mathscr{L} = (\nabla \times \overline{\boldsymbol{\mu}}^{-1} \cdot \nabla \times) - \omega^2 \overline{\boldsymbol{\epsilon}} \cdot$. Furthermore, f will be $\phi(\mathbf{r})$ for the scalar wave equation (36.1.1), while it will be $\mathbf{E}(\mathbf{r})$ in the case of vector wave equation for an electromagnetic system (36.1.2). The g on the right-hand side can represent Q in (36.1.1) or $i\omega\mathbf{J}(\mathbf{r})$ in (36.1.2).

36.1.2 Examples of Integral Equations

This course is replete with PDE's, but we have not come across too many integral equations. Therefore, we shall illustrate the derivation of some integral equations. Since the acoustic wave problem is homomorphic to the electromagnetic wave problem, we will illustrate the derivation of integral equation of scattering using acoustic wave equation.[1]

The surface integral equation method is rather popular in a number of applications, because it employs a homogeneous-medium Green's function which is simple in form, and the unknowns reside on a surface rather than in a volume. In this section, the surface integral equations[2] for scalar and will be studied first. Then, the volume integral equation will be discussed next.

36.1.3 Surface Integral Equations

In an integral equation, the unknown to be sought is embedded in an integral. An integral equation can be viewed as an operator equation as well, just as are differential equations. We shall see how such integral equations with only surface integrals are derived, using the scalar wave equation.

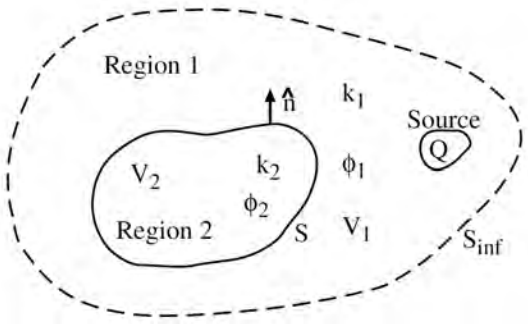

Figure 36.3: A two-region problem can be solved with a surface integral equation.

[1]The cases of electromagnetic wave equations can be found in *Chew, Waves and Fields in Inhomogeneous Media* [34].

[2]These are sometimes called boundary integral equations [211, 212].

Consider a scalar wave equation for a two-region problem as shown in Figure 36.3. In region 1, the governing equation for the total field is

$$(\nabla^2 + k_1^2)\,\phi_1(\mathbf{r}) = Q(\mathbf{r}), \tag{36.1.4}$$

For simplicity, we will assume that the scatterer is impenetrable, meaning that the field in region 2 is zero. Therefore, we need only define Green's functions for regions 1 to satisfy the following equations:

$$(\nabla^2 + k_1^2)\,g_1(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}'), \tag{36.1.5}$$

The derivation here is similar to the that of Huygens' principle. On multiplying Equation (36.1.1) by $g_1(\mathbf{r}, \mathbf{r}')$ and Equation (36.1.5) by $\phi_1(\mathbf{r})$, and then subtracting the two resultant equations, followed by integrating over region 1, we have, for $\mathbf{r}' \in V_1$,

$$\int_{V_1} dV\,[g_1(\mathbf{r}, \mathbf{r}')\nabla^2\phi_1(\mathbf{r}) - \phi_1(\mathbf{r})\nabla^2 g_1(\mathbf{r}, \mathbf{r}')] = \int_{V_1} dV\, g_1(\mathbf{r}, \mathbf{r}')Q(\mathbf{r}) + \phi_1(\mathbf{r}'), \quad \mathbf{r}' \in V_1. \tag{36.1.6}$$

Since $\nabla \cdot (g\nabla\phi - \phi\nabla g) = g\nabla^2\phi - \phi\nabla^2 g$, by applying Gauss' theorem, the volume integral on the left-hand side of (36.1.6) becomes a surface integral over the surface bounding V_1. Consequently,[3]

$$-\int_{S+S_{inf}} dS\,\hat{n} \cdot [g_1(\mathbf{r}, \mathbf{r}')\nabla\phi_1(\mathbf{r}) - \phi_1(\mathbf{r})\nabla g_1(\mathbf{r}, \mathbf{r}')] = -\phi_{inc}(\mathbf{r}') + \phi_1(\mathbf{r}'), \quad \mathbf{r}' \in V_1. \tag{36.1.7}$$

In the above, we have let

$$\phi_{inc}(\mathbf{r}') = -\int_{V_1} dV\, g_1(\mathbf{r}, \mathbf{r}')Q(\mathbf{r}), \tag{36.1.8}$$

since it is the incident field generated by the source $Q(\mathbf{r})$.

Note that up to this point, $g_1(\mathbf{r}, \mathbf{r}')$ is not explicitly specified, as long as it is a solution of (36.1.5). A simple choice for $g_1(\mathbf{r}, \mathbf{r}')$ that satisfies the radiation condition in region 1 is

$$g_1(\mathbf{r}, \mathbf{r}') = \frac{e^{ik_1|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r} - \mathbf{r}'|}, \tag{36.1.9}$$

It is the unbounded, homogeneous medium scalar Green's function. In this case, $\phi_{inc}(\mathbf{r})$ is the incident field generated by the source $Q(\mathbf{r})$ in the absence of the scatterer. Moreover, the

[3]The equality of the volume integral on the left-hand side of (36.1.6) and the surface integral on the left-hand side of (36.1.7) is also known as Green's theorem for some authors [82].

integral over S_{inf} vanishes when $S_{inf} \to \infty$ by virtue of the radiation condition.[4] Then, after swapping $\mathbf{r}$ and $\mathbf{r}'$, we have

$$\phi_1(\mathbf{r}) = \phi_{inc}(\mathbf{r}) - \int_S dS'\, \hat{n}' \cdot [g_1(\mathbf{r},\mathbf{r}')\nabla'\phi_1(\mathbf{r}') - \phi_1(\mathbf{r}')\nabla' g_1(\mathbf{r},\mathbf{r}')], \quad \mathbf{r} \in V_1. \tag{36.1.10}$$

But if $\mathbf{r}' \notin V_1$ in (36.1.6), the second term, $\phi_1(\mathbf{r})$, on the right-hand side of (36.1.6) would be zero, for $\mathbf{r}'$ would be in V_2 where the integration is not performed. Therefore, we can write (36.1.10) as

$$\left.\begin{array}{ll} \text{if } \mathbf{r} \in V_1, & \phi_1(\mathbf{r}) \\ \text{if } \mathbf{r} \in V_2, & 0 \end{array}\right\} = \phi_{inc}(\mathbf{r}) - \int_S dS'\, \hat{n}' \cdot [g_1(\mathbf{r},\mathbf{r}')\nabla'\phi_1(\mathbf{r}') - \phi_1(\mathbf{r}')\nabla' g_1(\mathbf{r},\mathbf{r}')]. \tag{36.1.11}$$

The above equation is evocative of Huygens' principle. It says that when the observation point $\mathbf{r}$ is in V_1, then the total field $\phi_1(\mathbf{r})$ consists of the incident field, $\phi_{inc}(\mathbf{r})$, and the contribution of field due to surface sources on S, which is the second term on the right-hand side of (36.1.11). But if the observation point is in V_2, then the surface sources on S generate a field that exactly cancels the incident field $\phi_{inc}(\mathbf{r})$, making the total field in region 2 zero. This fact is the core of the ***extinction theorem*** as shown in Figure 36.4 (see Born and Wolf 1980). These ideas were also discussed in the lecture on equivalence principles.

In (36.1.11), $\hat{n} \cdot \nabla\phi_1(\mathbf{r})$ and $\phi_1(\mathbf{r})$ act as surface sources. Moreover, they are impressed on S, creating a field in region 2 that cancels exactly the incident field in region 2 (see Figure 36.4).

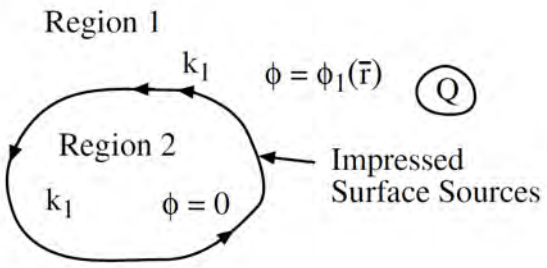

Figure 36.4: The illustration of the extinction theorem.

Applying the extinction theorem, integral equations can now be derived. So, using the lower parts of Equations (36.1.11), we have

$$\phi_{inc}(\mathbf{r}) = \int_S dS'\, \hat{n}' \cdot [g_1(\mathbf{r},\mathbf{r}')\nabla'\phi_1(\mathbf{r}') - \phi_1(\mathbf{r}')\nabla' g_1(\mathbf{r},\mathbf{r}')], \quad \mathbf{r} \in V_2, \tag{36.1.12}$$

The integral equations above still has two independent unknowns, ϕ_1 and $\hat{n} \cdot \nabla\phi_1$. Next, boundary conditions can be used to eliminate one of these two unknowns.

[4]The derivation here is similar to that for Huygens' principle in the previous lecture.

An acoustic scatterer which is impenetrable either has a hard surface boundary condition where normal velocity is zero, or it has soft surface where the pressure is zero (also called a pressure release surface). Since the force or the velocity of the particle is proportional to the $\nabla\phi$, a hard surface will have $\hat{n}\cdot\nabla\phi_1 = 0$, or a homogeneous Neumann boundary condition, while a soft surface will have $\phi_1 = 0$, a homogeneous Dirichlet boundary condition.

$$\phi_{inc}(\mathbf{r}) = \int_S dS'\,\hat{n}'\cdot[g_1(\mathbf{r},\mathbf{r}')\nabla'\phi_1(\mathbf{r}')], \quad \mathbf{r}\in V_2, \quad \text{soft boundary condition} \tag{36.1.13}$$

$$\phi_{inc}(\mathbf{r}) = -\int_S dS'\,\phi_1(\mathbf{r}')\nabla' g_1(\mathbf{r},\mathbf{r}'], \quad \mathbf{r}\in V_2, \quad \text{hard boundary condition} \tag{36.1.14}$$

The above are surface integral equations with the unknowns embedded in the integrals. They are $\hat{n}'\cdot\nabla'\phi(\mathbf{r}')$ and $\phi(\mathbf{r}')$ for (36.1.13) and (36.1.14), respectively.

More complicated surface integral equations (SIEs) for penetrable scatterers, as well as vector surface integral equations for the electromagnetics cases are derived in *Chew, Waves and Fields in Inhomogeneous Media* [34,213]. Also, there is another class of integral equations called volume integral equations (VIEs) [214]. They are also derived in [34].

Nevertheless, all linear integral equations can be unified under one notation:

$$\mathscr{L} f = g \tag{36.1.15}$$

where $\mathscr{L}$ is a linear operator. This is similar to the differential equation case expressed in (36.1.3). The difference is that here, the unknown f represents the source of the problem, while g is the incident field impinging on the scatterer or object. Furthermore, f does not need to satisfy any boundary condition, since the field radiated via the Green's function satisfies the radiation condition.

Several linear operator equations have been derived in the previous sections. They are all of the form

$$\mathscr{L} f = g \tag{36.1.16}$$

36.1.4 Function as a Vector

In the above, f is a functional vector which is the analogy of the vector $\mathbf{f}$ in matrix theory or linear algebra. In linear algebra, the vector $\mathbf{f}$ is of length N in an N dimensional space. It can be indexed by a set of countable index, say i, and we can described such a vector with N numbers such as $f_i, i = 1,\ldots,N$ explicitly. This is shown in Figure 36.5(a).

A function $f(x)$, however, can be thought of as being indexed by x in the 1D case. However, the index in this case is a continuum, and countably infinite. Hence, it corresponds to a vector of infinite dimension and it lives in an infinite dimensional space.[5]

To make such functions economical in storage, for instance, we replace the function $f(x)$ by its sampled values at N locations, such that $f(x_i), \quad i = 1,\ldots,N$. Then the values of the

[5]When these functions are square integrable implying finite "energy", these infinite dimensional spaces are called Hilbert spaces.

function in between the stored points $f(x_i)$ can be obtained by interpolation.[6] Therefore, a function vector $f(x)$, even though it is infinite dimensional, can be approximated by a finite length vector, $\mathbf{f}$. This concept is illustrated in Figure 36.5(b) and (c). This concept can be generalized to a function of 3D space $f(\mathbf{r})$. If $\mathbf{r}$ is sampled over a 3D volume, it can provide an index to a vector $f_i = f(\mathbf{r}_i)$, and hence, $f(\mathbf{r})$ can be thought of as a vector as well.

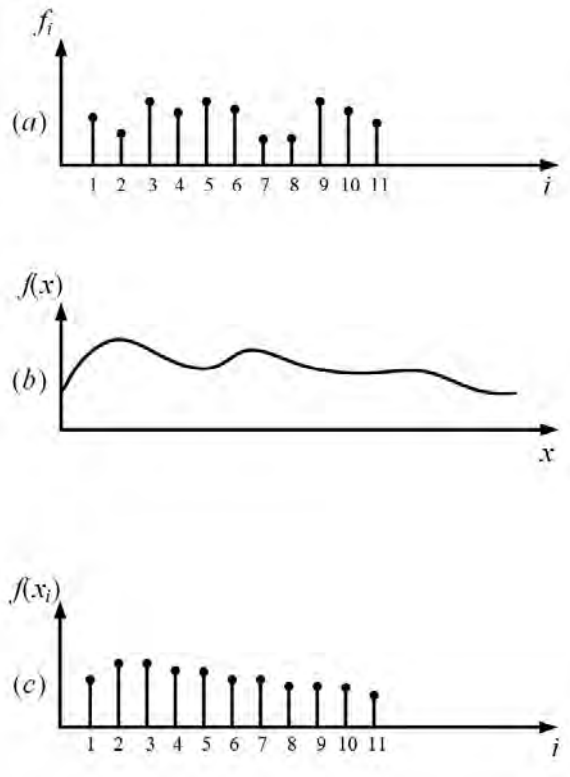

Figure 36.5: A function can be thought of as a vector.

36.1.5 Operator as a Map

Domain and Range Spaces

An operator like $\mathscr{L}$ above can be thought of as a map or a transformation. It maps a function f defined in a Hilbert space V to g defined in another Hilbert space W. Mathematically, this is written as

$$\mathscr{L} : V \to W \tag{36.1.17}$$

[6]This is in fact how special functions like $\sin(x)$, $\cos(x)$, $\exp(x)$, $J_n(x)$, $N_n(x)$, etc, are computed and stored in modern computers.

indicating that $\mathscr{L}$ is a map of vectors in the space V to vectors in the space W. Here, V is also called the ***domain space*** (or domain) of $\mathscr{L}$ while W is the ***range space*** (or range) of $\mathscr{L}$.

36.1.6 Approximating Operator Equations with Matrix Equations

36.2 Subspace Projection Methods

One main task of numerical method is first to approximate an operator equation $\mathscr{L}f = g$ by a matrix equation $\overline{\mathbf{L}} \cdot \mathbf{f} = \mathbf{g}$. To achieve the above, we first let

$$f \cong \sum_{n=1}^{N} a_n f_n = g \tag{36.2.1}$$

In the above, $f_n, n, \ldots, N$ are known functions called basis functions. Now, a_n's are the new unknowns to be sought. Also the above is an approximation, and the accuracy of the approximation depends very much on the original function f. A set of very popular basis functions are functions that form a piece-wise linear interpolation of the function from its nodes. These basis functions are shown in Figure 36.6.

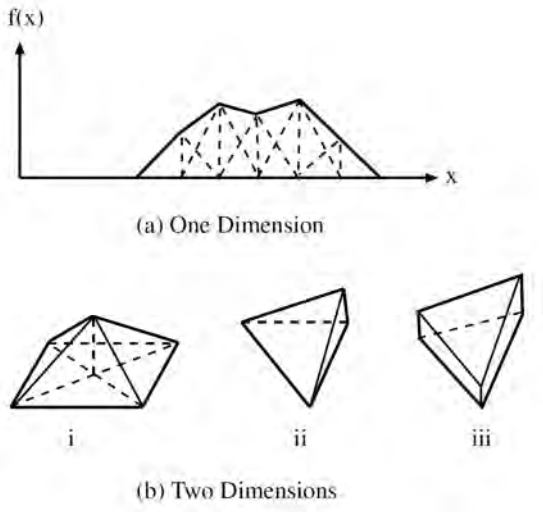

Figure 36.6: Examples of basis function in (a) one dimension, (b) two dimension. Each of these functions are define over a finite domain. Hence, they are also called sub-domain basis functions.

Upon substituting (36.2.1) into (36.1.16), we obtain

$$\sum_{n=1}^{N} a_n \mathscr{L} f_n = g \tag{36.2.2}$$

Then, upon multiplying (36.2.2) by w_m and integrate over the space that $w_m(\mathbf{r})$ is defined, then we have

$$\sum_{n=1}^{N} a_n \langle w_m, \mathscr{L} f_n \rangle = \langle w_m, g \rangle, m = 1, \ldots, N \tag{36.2.3}$$

In the above, the inner product is defined as

$$\langle f_1, f_2 \rangle = \int d\mathbf{r} f_1(\mathbf{r}) f_2(\mathbf{r}) \tag{36.2.4}$$

where the integration is over the support of the functions, or the space over which the functions are defined.[7] For PDEs these functions are defined over a 3D coordinate space, while in SIEs, these functions are defined over a surface. In a 1D problems, these functions are defined over a 1D coordinate space.

Dual Spaces

The functions $w_m, m = 1, \ldots, N$ is known as the weighting functions or testing functions. The testing functions should be chosen so that they can approximate well a function that lives in the range space W of the operator $\mathscr{L}$. Such set of testing functions lives in the ***dual space*** of the range space. For example, if f_r lives in the range space of the operator $\mathscr{L}$, the set of function f_d, such that the inner product $\langle f_d, f_r \rangle$ exists, forms the dual space of W.

Matrix and Vector Representations

The above is a matrix equation of the form

$$\overline{\mathbf{L}} \cdot \mathbf{a} = \mathbf{g} \tag{36.2.5}$$

where

$$\begin{aligned} \left[\overline{\mathbf{L}}\right]_{mn} &= \langle w_m, \mathscr{L} f_n \rangle \\ [\mathbf{a}]_n = a_n, [\mathbf{g}]_m &= \langle w_m, g \rangle \end{aligned} \tag{36.2.6}$$

What has effectively happened here is that given an operator $\mathscr{L}$ that maps a function that lives in an infinite dimensional Hilbert space V, to another function that lives in another infinite dimensional Hilbert space W, via the operator equation $\mathscr{L} f = g$, we have approximated the Hilbert spaces with finite dimensional spaces (subspaces), and finally, obtain a finite dimensional matrix equation that is the representation of the original infinite dimensional operator equation. This is the spirit of the subspace projection method.

In the above, $\overline{\mathbf{L}}$ is the matrix representation of the operator $\mathscr{L}$ in the subspaces, and $\mathbf{a}$ and $\mathbf{g}$ are the vector representations of f and g, respectively, in their respective subspaces.

When such a method is applied to integral equations, it is usually called the method of moments (MOM). (Surface integral equations are also called boundary integral equations (BIEs) in other fields [212]). When finite discrete basis are used to represent the surface unknowns, it is also called the boundary element method (BEM) [215]. But when this method is applied to solve PDEs, it is called the finite element method (FEM) [216–219], which is a rather popular method due to its simplicity.

[7] This is known as the reaction inner product [34, 45, 121]. As oppose to most math and physics literature, the energy inner product is used [121] where $\langle f_1, f_2 \rangle = \int d\mathbf{r} f_1^*(\mathbf{r}) f_2(\mathbf{r})$.

36.2.1 Mesh Generation

In order to approximate a function defined on an arbitrary shaped surface or volume by a finite sum of basis functions, it is best to mesh (tessellate or discretize) the surface and volume by meshes. In 2D, all shapes can be tessellated by unions of triangles, while a 3D volume can be meshed (tessellated) by unions of tetrahedrons. Such meshes are used not only in CEM, but in other fields such as solid mechanics. Hence, there are many "solid modeling" commercial software available to generate sophisticated meshes.

When a surface is curved, or of arbitrary shape, it can be meshed by union of triangles as shown in Figure 36.7. When a volume is of arbitrary shape or a volume is around an arbitrary shape object, it can be meshed by tetrahedrons as shown in Figure 36.8. Then basis functions as used in (36.2.1) are defined to interpolate the field between nodal values or values defined on the edges of a triangle or a tetrahedron.

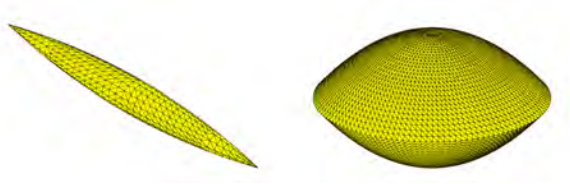

Figure 36.7: An arbitrary surface can be meshed by a union of triangles.

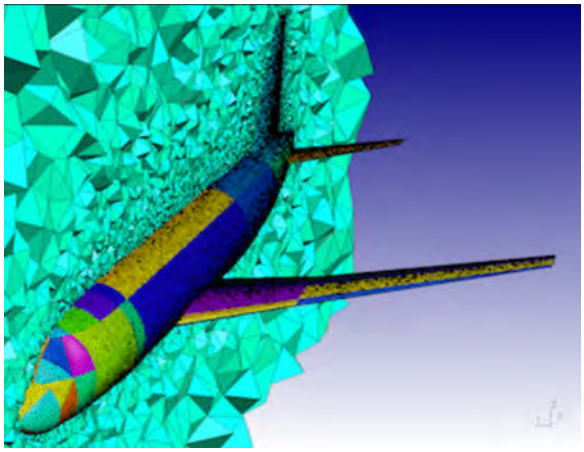

Figure 36.8: A volume region can be meshed by a union of tetrahedra. But the surface of the aircraft is meshed with a union of trianlges (courtesy of gmsh.info).

36.2.2 Differential Equation Solvers versus Integral Equation Solvers

As have been shown, the two classes of numerical solvers for Maxwell's equations consist of differential equation solvers and integral equation solvers. Differential equation solvers are

generally easier to implement. As shall be shown in the next lecture, they can also be easily implemented using finite difference solver. The unknowns in a differential equation solver are the fields. The fields permeate all of space, and hence, the unknowns are volumetrically distributed. When the fields are digitized by representing them by their point values in space, they require a large number of unknowns to represent. The plus side is that the matrix system associated with a differential equation solver is usually sparse, requiring less storage and less time to solve.

As has been shown, integral equation solvers are formulated using Green's functions. That is integral equations are derived from Maxwell's equations using Green's function, where the unknowns now are surface sources such as surface electric and magnetic currents. Therefore, the unknowns are generally smaller, living only on the surface of a scatterer (or they occupy a smaller part of space). Hence, they can be approximated by a smaller set of unknowns. Thus, the matrix systems generally are smaller. Once the currents are found, then the fields they generate can also be computed.

Since the derivation of integral equations requires the use of Green's functions, they are in general singular when $\mathbf{r} = \mathbf{r}'$, or when the observation point (observation point) $\mathbf{r}$ and the source point $\mathbf{r}'$ coincide. Care has to be taken to discretize the integral equations. Moreover, a Green's function connects every current source point on the surface of a scatterer with every other source points yielding a dense matrix system. But fast methods have been developed to solve such dense matrix systems [9].

36.3 Solving Matrix Equation by Optimization

When a matrix system get exceedingly large, it is preferable that a direct inversion of the matrix equation not performed. Direct inversions (e.g., using Gaussian elimination [220] or Kramer's rule [221]) have computational complexity[8] of $O(N^3)$, and requiring storage of $O(N^2)$. Hence, when N is large, other methods have to be sought.

To this end, it is better to convert the solving of a matrix equation into an optimization problem. These methods can be designed so that a much larger system can be solved with an existing digital computer. Optimization problem results in finding the stationary point of a functional.[9] First, we will figure out how to find such a functional.

Consider a matrix equation given by

$$\overline{\mathbf{L}} \cdot \mathbf{f} = \mathbf{g} \tag{36.3.1}$$

For simplicity, we consider $\overline{\mathbf{L}}$ as a symmetric matrix.[10] Then the corresponding functional is

$$I = \mathbf{f}^t \cdot \overline{\mathbf{L}} \cdot \mathbf{f} - 2\mathbf{f}^t \cdot \mathbf{g} \tag{36.3.2}$$

Such a functional is called a quadratic functional because it is analogous to $I = Lx^2 - 2xg$, which is quadratic, in its simplest 1D rendition.

[8]The scaling of computer time with respect to the number of unknowns (degrees of freedom) is known in the computer parlance as computational complexity.

[9]Functional is usually defined as a function of a function [34, 44]. Here, we include a function of a vector to be a functional as well.

[10]Functional for the asymmetric case can be found in *Chew, Waves and Fields in Inhomogeneous Media*, Chapter 5 [34].

Taking the first variation with respect to $\mathbf{f}$, namely, we let $\mathbf{f} = \mathbf{f}_o + \delta\mathbf{f}$. Then we substitute this into the above, and collect the leading order and first order terms. Then we find the first order approximation of the functional I as

$$\delta I = \delta\mathbf{f}^t \cdot \overline{\mathbf{L}} \cdot \mathbf{f}_o + \mathbf{f}_o^t \cdot \overline{\mathbf{L}} \cdot \delta\mathbf{f} - 2\delta\mathbf{f}^t \cdot \mathbf{g} \tag{36.3.3}$$

If $\overline{\mathbf{L}}$ is symmetric, the first two terms are the same, and the above becomes

$$\delta I = 2\delta\mathbf{f}^t \cdot \overline{\mathbf{L}} \cdot \mathbf{f}_o - 2\delta\mathbf{f}^t \cdot \mathbf{g} \tag{36.3.4}$$

For $\mathbf{f}_o$ to be the optimal point or the stationary point, then its first variation has to be zero, or that $\delta I = 0$. Thus we conclude that at the optimal point (or the stationary point),

$$\overline{\mathbf{L}} \cdot \mathbf{f}_o = \mathbf{g} \tag{36.3.5}$$

Hence, the optimal point to the functional I in (36.3.2) is the solution to (36.3.1) or (36.3.5).

36.3.1 Gradient of a Functional

The above method, when applied to an infinite dimensional Hilbert space problem, is called variational method, but the main ideas are similar. The wonderful idea about such a method is that instead of doing direct inversion of a matrix system (which is expensive), one can search for the optimal point or stationary point of the quadratic functional using gradient search or gradient descent methods or some optimization method.

It turns out that the gradient of a quadratic functional can be found quite easily. Also it is cheaper to compute the gradient of a functional than to find the inverse of a matrix operator. To do this, it is better to write out functional using index (or indicial, or Einstein) notation [222]. In this notation, the functional first variation δI in (36.3.4) becomes

$$\delta I = 2\delta f_j L_{ij} f_i - 2\delta f_j g_j \tag{36.3.6}$$

Also, in this notation, the summation symbol is dropped, and summations over repeated indices are implied. In the above, we neglect to distinguish between $\mathbf{f}_o$ and $\mathbf{f}$. It is implied that $\mathbf{f}$ represents the optimal point. In this notation, it is easier to see what a functional derivative is. We can differentiate the above with respect to f_j easily to arrive at

$$\frac{\partial I}{\partial f_j} = 2L_{ij} f_i - 2g_j \tag{36.3.7}$$

Notice that the remaining equation has one index j remaining in index notation, meaning that it is a vector equation. We can reconstitute the above using our more familiar matrix notation that

$$\frac{\delta I}{\delta \mathbf{f}} = \nabla_{\mathbf{f}} I = 2\overline{\mathbf{L}} \cdot \mathbf{f} - 2\mathbf{g} \tag{36.3.8}$$

The left-hand side is a notation for the functional derivative or the gradient of a functional in a multi-dimensional space whichi is a vector obviated by indicial notation. And the right-hand

side is the expression for calculating this gradient. One needs only to perform a matrix-vector product to find this gradient. Hence, the computational complexity of finding this gradient is $O(N^2)$ at worst if $\overline{\mathbf{L}}$ is a dense matrix, and as low as $O(N)$ if $\overline{\mathbf{L}}$ is a sparse matrix. In a gradient search method, such a gradient is calculated repeatedly until the optimal point is found. Such methods are called iterative methods.

If the optimal point can be found in N_{iter} iterations, then the CPU time scales as $N_{iter}N^{\alpha}$ where $1 < \alpha < 2$. There is a clever gradient search algorithm, called the ***conjugate gradient method*** that can find the optimal point in N_{iter} in exact arithmetics. In many gradient search methods, $N_{\text{iter}} \ll N$. The total solution time or solve time which is $N_{iter}N^{\alpha} \ll NN^{\alpha} \ll N^3$, resulting in great savings in computer time.

What is more important is that this method does not require the storage of the matrix $\overline{\mathbf{L}}$, but a computer code that produces the vector $\mathbf{g}_o = \overline{\mathbf{L}} \cdot \mathbf{f}$ as an output, with $\mathbf{f}$ as an input. Both $\mathbf{f}$ and $\mathbf{g}_o$ require only $O(N)$ memory storage. Such methods are called matrix-free methods. Even when $\overline{\mathbf{L}}$ is a dense matrix, which is the case if it is the matrix representation of matrix representation of some Green's function, fast methods now exist to perform the dense matrix-vector product in $O(N \log N)$ operations.[11]

The value I is also called the cost function, and its minimum is sought in the seeking of the solution by gradient search methods. Detail discussion of these methods is given in [223]. Figure 36.9 shows the contour plot of a cost function in 2D. When the condition number[12] of the matrix $\overline{\mathbf{L}}$ is large (implying that the matrix is ill-conditioned), the contour plot will resemble a deep valley. And hence, the gradient search method will tend to zig-zag along the way as it finds the solution. Therefore, convergence is slow for matrices with large condition numbers

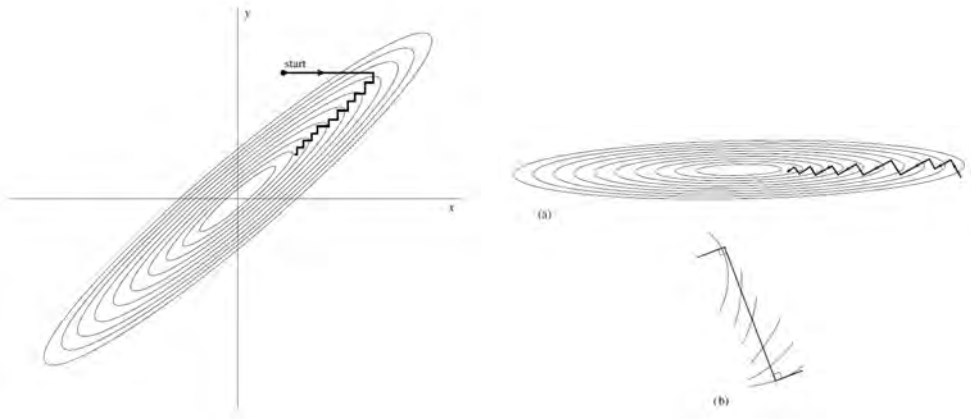

Figure 36.9: Plot of a 2D cost function, $I(x, y)$ for an ill-conditioned system (courtesy of Numerical Recipe [223]). A higher dimensional plot of this cost function will be difficult.

Figure 36.10 shows a cartoon picture in 2D of the histories of different search paths from a

[11] Chew et al, *Fast and Efficient Algorithms in CEM* [9].

[12] This is the ratio of the largest eigenvalue of the matrix to its smallest eigenvalue.

machine-learning example where a cost functional similar to I has to be minimized. Finding the optimal point or the minimum point of a general functional is still a hot topic of research: it is important in artificial intelligence.

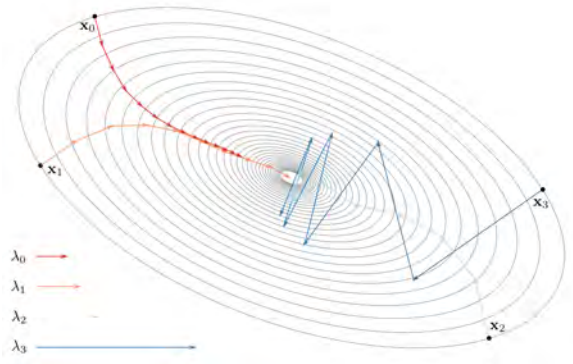

Figure 36.10: Gradient search or gradient descent method is finding an optimal point (courtesy of Y. Ioannou: https://blog.yani.io/sgd/).