Simulation Electrodynamics

Rasmus Bruhn Nielsen

1 Non-Responsive Electrodynamics

The goal of these notes will be to develop a way to simulate electrodynamics. We will approximate Maxwell's equations to be able to simulate them with focus on speed of the simulation as well as saving memory. In this chapter we want to focus on non-responsive electrodynamics, that is electrodynamics where no charges respond to the induced electric or magnetic fields. Later on we will try to expand the model to allow current flow and charge displacement in response to electromagnetic fields (metals), polarization of materials (dielectrics) and magnetization (dia-, para- and ferromagnets). Throughout the notes we will be using the text book "Introduction to Electrodynamics" by David J. Griffiths, we will be referring to an equation from this as (Griffiths, [Number]).

1.1 The Theory

To start off with we have Maxwell's equations (Griffiths, 7.40)

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t}
\end{aligned} \tag{1}$$

Where $\mathbf{E}, \mathbf{B}$ are the electric and magnetic fields, $\rho, \mathbf{J}$ are the charge and current density and ϵ_0, μ_0 are the permittivity and permeability of free space with the relation $\frac{1}{\epsilon_0 \mu_0} = c^2$ where c is the speed of light.

The big problem with these equations are that they are 6 functions which are determined by first order partial differential equations mixing all the functions together. If we instead use potentials we can simplify this a lot. Then we can turn the problem into finding 4 functions determined by independent second order partial differential equations. This will be much easier to simulate as we don't need to take into account the effects of the functions on each other and we go from 6 to 4 functions which will save time. The potentials $V, \mathbf{A}$ are defined by:

$$\begin{aligned}
\mathbf{E} &= -\nabla V - \frac{\partial \mathbf{A}}{\partial t} \\
\mathbf{B} &= \nabla \times \mathbf{A}
\end{aligned} \tag{2}$$

Using potentials and the Lorenz gauge transformation, Maxwell's equations become (Griffiths, 10.16)

$$\begin{aligned}
\frac{1}{c}\Box^2 V &= -\mu_0 c \rho \\
\Box^2 \mathbf{A} &= -\mu_0 \mathbf{J}
\end{aligned} \tag{3}$$

Where $\Box^2 \equiv \nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}$

We can write this equation event shorter if we define:

$$\begin{aligned} A^\mu &= \left(\frac{1}{c}V, A_x, A_y, A_z\right) \\ J^\mu &= (c\rho, J_x, J_y, J_z) \end{aligned} \tag{4}$$

Here A^μ and J^μ are just 4 dimensional vectors. You get a component of one of these vectors by inserting a number instead of μ. So $A^0 = \frac{1}{c}V$ and $A^2 = A_y$. This notation is reminiscent of special relativity where this is 4-vector notation, but here we will not consider special relativity and it is just a normal vector. Then our defining differential equation becomes:

$$\Box^2 A^\mu = -\mu_0 J^\mu \tag{5}$$

It is the exact same 4 equations just written in a different notation. Now we have achieved what we wanted, we have now obtained 4 independent second order partial differential equations. So now we just need to simulate this much simpler equation and then we can calculate the electromagnetic fields using Eq. 2.

1.2 The Model

We will model our system as a 3D grid with a finite resolution along each axis, and we will use Cartesian coordinates. The resolution along the axis will be given as $\Delta x, \Delta y, \Delta z$. For shorthand notation we will write $\Delta\mathbf{x} \equiv (\Delta x, \Delta y, \Delta z)$ and $\mathbf{x} \equiv (x, y, z)$. We also assume that the boundary conditions are predefined and not cyclic, although cyclic boundary conditions does only require small changes to the result we will reach. This means that for a function $f(x, y, z, t)$ of 3 spacial variables and 1 time variable, we sample it as $f(n_x\Delta x, n_y\Delta y, n_z\Delta z, t)$ with the value $n_i \in \mathbb{N}$ and $0 \leq n_i < N_i$ with $N_i \in \mathbb{N}$. Here $\mathbf{N} \equiv (N_x, N_y, N_z)$ is the resolution of the model and $\mathbf{n} \equiv (n_x, n_y, n_z)$ determines the position. Again we introduce shorthand notation $f(\mathbf{x_n}, t) \equiv f(n_x\Delta x, n_y\Delta y, n_z\Delta z, t)$ where $\mathbf{x_n} \equiv n_x\Delta x\hat{\mathbf{x}} + n_y\Delta y\hat{\mathbf{y}} + n_z\Delta z\hat{\mathbf{z}}$. Sometimes we will also use $x = x^1, y = x^2, z = x^3$. We will also model this in discrete intervals with the length of the intervals being Δt.

For our model to be good we need that the change of any function f in a time interval Δt is small, and we need that the functions we will simulate don't change much between neighbouring points. For the first condition to be correct we need that $\frac{\Delta x_i}{\Delta t} >> c$ since then changes will take many frames to propagate and thus it should happen slowly. Basically Δt needs to be small enough. And for the second condition to be true we must have that the distance between points Δx_i is small enough. What we are saying here is just that the resolution of our problem should be small enough for a given problem, which is something that needs to be true for all simulations.

We will also require for any function f that:

$$\left|\sum_{n=3}^{\infty} \frac{1}{n!}\frac{\partial^n f}{\partial x^n}(\Delta x)^n\right| << \left|\frac{\partial f}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(\Delta x)^2\right| \tag{6}$$

For some variable x and the resolution in that variable Δx, note that x could also be time here. We assume this so that we can Taylor expand functions to second order later on.

Partial Derivatives

Before we can go on we first needs to do some approximations to the partial derivatives. We will not consider partial derivatives of orders greater that 2 since we will not need these. Since our functions are no longer continuous we can no longer calculate the partial derivatives of our functions. Therefor we will need to find an approximate way to get the derivatives. To do this we will use Taylor expansion.

Assume we have a function f, it is a function of some variable x with a resolution Δx. The function could also depend on more variables but we are only interested in x since we only consider partial derivatives. Now Taylor expand the function around some value $x = x_0$

$$f(x + \delta x) \approx f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x=x_0} \delta x + \frac{1}{2} \left.\frac{\partial^2 f}{\partial x^2}\right|_{x=x_0} (\delta x)^2 \tag{7}$$

Using this we see that

$$\begin{aligned} \left.\frac{\partial f}{\partial x}\right|_{x=x_0} &\approx \frac{1}{2\Delta x}(f(x_0 + \Delta x) - f(x_0 - \Delta x)) \\ \left.\frac{\partial^2 f}{\partial x^2}\right|_{x=x_0} &\approx \frac{1}{(\Delta x)^2}(f(x_0 + \Delta x) + f(x_0 - \Delta x) - 2f(x_0)) \end{aligned} \tag{8}$$

1.3 Solving The Problem

We are now ready to solve our problem to find a way to simulate electromagnetic fields in the non-responsive electrodynamics case. First we make some assumptions.

Assumptions

We assume that the generalized current vector J^μ is a predefined function, it can change with time but it must be predefined. Otherwise it would be responsive. We also assume that we have some starting conditions. So we assume given

$$\begin{aligned} &J^\mu(\mathbf{x}, t) \\ &A^\mu(\mathbf{x}, t_0) \\ &\left.\frac{\partial A^\mu}{\partial t}\right|_{t=t_0} \end{aligned} \tag{9}$$

For all $\mathbf{x}$ and some initial time t_0.

We will also assume the boundary condition that $A^\mu = 0$ outside our model.

Approximating

In the following if no space coordinate is given for a function then $\mathbf{x}$ is implicitly included. From Eq. 5 we get

$$\frac{\partial^2 A^\mu}{\partial t^2} = c^2\big(\nabla^2 A^\mu + \mu_0 J^\mu(t)\big) \tag{10}$$

If we now Taylor expand A^μ at the time $t_0 + \frac{1}{2}\Delta t$ then we get

$$\begin{aligned} A^\mu\left(t_0 + \frac{1}{2}\Delta t + \delta t\right) &= A^\mu\left(t_0 + \frac{1}{2}\Delta t\right) + \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t}\delta t + \frac{1}{2}\frac{\partial^2 A^\mu}{\partial t^2}\bigg|_{t=t_0+\frac{1}{2}\Delta t}(\delta t)^2 \\ &= A^\mu\left(t_0 + \frac{1}{2}\Delta t\right) + \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t}\delta t + \frac{1}{2}c^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right)(\delta t)^2 \end{aligned} \tag{11}$$

And by differentiating Eq. 11 we get

$$\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t+\delta t} = \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t} + c^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right)\delta t \tag{12}$$

Now we need $A^\mu\big(t_0 + \frac{1}{2}\Delta t + \delta t\big)$ and $\frac{\partial A^\mu}{\partial t}\Big|_{t=t_0+\frac{1}{2}\Delta t}$ in terms of the starting conditions from Eq. 9. By inserting $\delta t = -\frac{1}{2}\Delta t$ in Eq. 12 we get

$$\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0} = \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t} - \frac{1}{2}c^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right)\Delta t \tag{13}$$

So

$$\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\frac{1}{2}\Delta t} = \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0} + \frac{1}{2}c^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right)\Delta t \tag{14}$$

We can insert this into Eq. 11 to get

$$A^\mu\left(t_0 + \frac{1}{2}\Delta t + \delta t\right) = A^\mu\left(t_0 + \frac{1}{2}\Delta t\right) + \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0}\delta t + \frac{1}{2}c^2\delta t(\delta t + \Delta t)\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right) \tag{15}$$

Setting $\delta t = -\frac{1}{2}\Delta t$ in Eq. 15 we get

$$A^\mu(t_0) = A^\mu\left(t_0 + \frac{1}{2}\Delta t\right) - \frac{1}{2}\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0}\Delta t - \frac{1}{8}c^2(\Delta t)^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} + \mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right)\right) \tag{16}$$

So

$$A^\mu\left(t_0 + \frac{1}{2}\Delta t\right) - \frac{1}{8}c^2(\Delta t)^2\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t} = A^\mu(t_0) + \frac{1}{2}\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0}\Delta t + \frac{1}{8}c^2(\Delta t)^2\mu_0 J^\mu\left(t_0 + \frac{1}{2}\Delta t\right) \equiv R^\mu(t_0) \tag{17}$$

This is an implicit equation for $A^\mu\left(t_0+\frac{1}{2}\Delta t\right)$ which we will need to solve. Once we have solved it we can use that

$$\frac{1}{8}c^2(\Delta t)^2\left(\nabla^2 A^\mu\bigg|_{t=t_0+\frac{1}{2}\Delta t}+\mu_0 J^\mu\left(t_0+\frac{1}{2}\Delta t\right)\right)=A^\mu\left(t_0+\frac{1}{2}\Delta t\right)-A^\mu(t_0)-\frac{1}{2}\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0}\Delta t \tag{18}$$

Then by inserting $\delta t=\frac{1}{2}\Delta t$ and Eq. 18 into Eq. 15 and Eq. 12 we get

$$\begin{aligned} A^\mu(t_0+\Delta t) &= 4A^\mu\left(t_0+\frac{1}{2}\Delta t\right)-3A^\mu(t_0)-\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0}\Delta t \\ \frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0+\Delta t} &= \frac{8}{\Delta t}\left(A^\mu\left(t_0+\frac{1}{2}\Delta t\right)-A^\mu(t_0)\right)-3\frac{\partial A^\mu}{\partial t}\bigg|_{t=t_0} \end{aligned} \tag{19}$$

Calculating The Result

Now we have a theoretical way to determine $A^\mu(t_0+\Delta t)$ form and $\frac{\partial A^\mu}{\partial t}\Big|_{t=t_0+\Delta t}$ using $A^\mu(t_0)$ for Eq. 19 and Eq. 17. Although we have these equations, we still do not have any way to calculate it since we don't know how to calculate $\nabla^2 A^\mu$. So now we need to use the fact that we are using Cartesian coordinates. Until now it has been completely general but now we will write the laplacian in Cartesian coordinates

$$\nabla^2 A^\mu=\frac{\partial^2 A^\mu}{\partial x^2}+\frac{\partial^2 A^\mu}{\partial y^2}+\frac{\partial^2 A^\mu}{\partial z^2} \tag{20}$$

Using Eq. 8 we can approximate this as

$$\nabla^2 A^\mu(\mathbf{x})=\sum_{i\in\{x,y,z\}}\left(\frac{1}{(\Delta x_i)^2}\left(A^\mu(\mathbf{x}+\Delta x_i\hat{\mathbf{x}}_\mathbf{i})+A^\mu(\mathbf{x}-\Delta x_i\hat{\mathbf{x}}_\mathbf{i})-2A^\mu(\mathbf{x})\right)\right) \tag{21}$$

Where everything is evaluated at the same time. Inserting this into Eq. 17 we get

$$A^\mu(\mathbf{x})-\frac{1}{8}c^2(\Delta t)^2\sum_{i\in\{x,y,z\}}\left(\frac{1}{(\Delta x_i)^2}\left(A^\mu(\mathbf{x}+\Delta x_i\hat{\mathbf{x}}_\mathbf{i})+A^\mu(\mathbf{x}-\Delta x_i\hat{\mathbf{x}}_\mathbf{i})-2A^\mu\left(\mathbf{x},t_0+\frac{1}{2}\Delta t\right)\right)\right)=R^\mu(\mathbf{x},t_0) \tag{22}$$

Where everything on the left side is evaluated at the time $t_0+\frac{1}{2}\Delta t$.

This is a beastly set of equations, if we define $V\equiv N_xN_yN_z$ then we have V variables with V coupled equations. For a simulation we will expect V to be between 10^6 and 10^9 so we have a lot of equations. Luckily they are linear, so we can solve the problem using linear algebra. We could also try to solve it using other methods but linear algebra is easy to use and it is very widely used, so it is very optimized.

To use linear algebra we need to be able to put all values of A^μ into a vector. This can be done in many different ways but we choose to define $\mathbf{A}^\mu$ such that

$$A^\mu(n_x\Delta x,n_y\Delta y,n_z\Delta z)=(\mathbf{A}^\mu)_{n_x+n_yN_x+n_zN_xN_y} \tag{23}$$

Now we can define the matrix representing the laplacian

$$(\mathbf{M}^\mu)_{nm} = \begin{cases} -2\sum_{i\in\{x,y,z\}} \frac{1}{(\Delta x_i)^2} & \text{for } m = n \\ \frac{1}{(\Delta x)^2} & \text{for } (m = n+1 \text{ and } n_x \neq N_x - 1) \text{ or } (m = n-1 \text{ and } n_x \neq 0) \\ \frac{1}{(\Delta y)^2} & \text{for } (m = n+N_x \text{ and } n_y \neq N_y - 1) \text{ or } (m = n-N_x \text{ and } n_y \neq 0) \\ \frac{1}{(\Delta z)^2} & \text{for } (m = n+N_xN_y \text{ and } n_z \neq N_z - 1) \text{ or } (m = n-N_xN_y \text{ and } n_z \neq 0) \\ 0 & \text{else} \end{cases} \tag{24}$$

For $n = n_x + n_yN_x + n_zN_xN_y$.

If we also define the vector $\mathbf{S}^\mu$ as s^μ ordered in the same way as $\mathbf{A}^\mu$ then we can rewrite our set of equations to

$$\left(\mathbf{I} - \frac{1}{8}c^2(\Delta t)^2\mathbf{M}^\mu\right)\mathbf{A}^\mu = \mathbf{R}^\mu \tag{25}$$

Really what we have here is 4 matrix equations, one for each value of μ. Now we can also notice that the demand that $\frac{\Delta x_i}{\Delta t} >> c$ is equivalent to saying that $\left|\left(\frac{1}{8}c^2(\Delta t)^2\mathbf{M}^\mu\right)_{nm}\right| << 1$ or $\left(\mathbf{I} - \frac{1}{8}c^2(\Delta t)^2\mathbf{M}^\mu\right) \approx \mathbf{I}$.

Now we just needs to solve this matrix equation. There are however 2 problems. First lets look at the size of the matrix, it is of size V^2. If we are to store this matrix on a computer then for a small system with $V = 10^6$ and 32-bit floating point values we will still need 4000 gigabytes of memory. This is completely impossible, but luckily there is a way around this. We notice that the matrix in question mostly consists of 0's and therefor we can use something known as a sparse matrix. This is a matrix where you only save the non-zero elements. Then the memory usage is of order V and not V^2 which means that we can save it in the memory. The other problem is that if we try to use a linear solver we will have to wait for a long time to simulate anything. The reason for this is that the speed of such a solver is of the order of the size of the matrix, so V^2. Since we need to solve this for every single time interval, the simulation will never end. So we must find an approximation. If we rearrange we get

$$\mathbf{A}^\mu = \frac{1}{8}c^2(\Delta t)^2\mathbf{M}^\mu\mathbf{A}^\mu + \mathbf{R}^\mu \tag{26}$$

We will now introduce a series $(\mathbf{A}^\mu_\mathbf{0}, \mathbf{A}^\mu_\mathbf{1}, ...)$ defined by

$$\begin{aligned} \mathbf{A}^\mu_\mathbf{n+1} &= \frac{1}{8}c^2(\Delta t)^2\mathbf{M}^\mu\mathbf{A}^\mu_\mathbf{n} + \mathbf{R}^\mu \\ \mathbf{A}^\mu_\mathbf{0}\left(t_0 + \frac{1}{2}\Delta t\right) &= \mathbf{A}^\mu(t_0) + \frac{1}{2}\Delta t\frac{\partial \mathbf{A}^\mu}{\partial t}\bigg|_{t=t_0} \end{aligned} \tag{27}$$

We notice that if the series $\{\mathbf{A}^\mu_\mathbf{n}\}$ converges then the limit is

$$\lim_{n\to\infty} \mathbf{A}^{\mu}_{\mathbf{n}}\left(t_0 + \frac{1}{2}\Delta t\right) = \mathbf{A}^{\mu}\left(t_0 + \frac{1}{2}\Delta t\right) \tag{28}$$

And from testing in Python we see that it does converge as long as $\frac{\Delta x_i}{\Delta t}$ is large enough. Using the method the speed of the calculation will be of order V. It will not be exact but if changes happen slowly enough then it will give the answer to a high precision within just a few iterations.

1.4 Non-Responsive Electrostatics

Now we have a complete model and we can implement it in Python. It is fairly easy to do since it is just some linear algebra. But when trying it we get nonsense results, as an example a point charge will create an oscillating field. This makes no sense. The reason for this is actually not a numerical error but arises because our starting conditions are non-physical. For the point charge, the starting condition was just $\mathbf{A}^{\mu} = \mathbf{0}$. But we can't just make a charge appear in empty space. To fix this we must first calculate the starting condition. What we will do is assume that the system is frozen in time at $t = 0$ and calculate the starting condition here, then we will use this result as the starting condition for the simulation. This means that we have to perform an electrostatics problem now.

Changing The Model

For electrostatics we have $\frac{\partial^2 A^{\mu}}{\partial t^2} = 0$, then Eq. 10 becomes

$$\nabla^2 A^{\mu} = -\mu_0 J^{\mu} \tag{29}$$

Representing this as a matrix equation we get

$$\mathbf{M}^{\mu}\mathbf{A}^{\mu} = -\mu_0 \mathbf{J}^{\mu} \tag{30}$$

Again using a linear solver will be very slow but may be possible here since we only need to solve it once, and not for every time interval. If it is too slow we can make the same kind of approximating as before

$$\begin{aligned} &\mathbf{A}^{\mu}_{\mathbf{n+1}} = (\mathbf{I} + k\mathbf{M}^{\mu})\mathbf{A}^{\mu}_{\mathbf{n}} + k\mu_0\mathbf{J}^{\mu} \\ &\mathbf{A}^{\mu}_{\mathbf{0}} = \mathbf{0} \end{aligned} \tag{31}$$

For any number k. This series should approach the correct result. We can also define a different $\mathbf{A}^{\mu}_{\mathbf{0}}$ if we have a better guess, this may make the convergence faster.