# Modelling and Design of a Quadrupole Ion Trap

Lara V. Scavuzzo
*ETSETB - Universitat Politècnica de Catalunya*
(Dated: May 23, 2016)

A quadrupole ion trap (also known as Paul trap) is a device that confines charged particles using dynamic electric fields produced by four voltage-varying electrodes. The aim of this project is to simulate numerically the movement of such ions and therefore design the optimal ion trap. In order to do so, a number of simulations were made from different approaches: by direct discretization of the differential equations with the finite element method (FEM), and also by discretizing the integral equation using the method of moments (MoM), both in 2D and 3D. The voltage applied to the electrodes presents a harmonic variation, which takes place at a frequency that is low enough to consider a quasi-static problem, fact that simplifies the resolution.

## I. FIRST APPROACH: MATHEMATICAL ANALYSIS

The first step is to set out the equations that must be solved in order to obtain the potential function at every point in space. The 2D case was adressed by means of both MoM and FEM, but the latter method was not numerically compatible with the 3D problem. A discretization must be made in both cases. However, in the FEM case this is done through all space, while MoM uses only a surface discretization, over the points where we impose boudary conditions. The number of unknowns was already extremely high in 2D for FEM, fact that was compensated by the resulting sparse matrix, which made the linear system resolution much simpler. However in the 3D case, MoM showed to be more suitable because of the surface vs volume discretization, which reduces considerably the number of unknowns. In this section, a brief summary of the principles of both methods will be presented.

### A. 2D Finite element method

A matlab toolbox [1] was used in order to produce the necessary geometry and meshing, as well as solving Laplace's equation $\nabla^2\Phi(\vec{r}) = 0$ outside the plate's geometry (in this case a certain width had to be considered). The adaptive meshing was done automatically by the toolbox. The geometries that were tested for this case were two parallel plates, and four plates forming a sort of open rectangle.

It is important to set the appropriate boundary conditions. The discretization is made inside a "box" whose dimensions we can choose (we cannot calculate the potential in literally all points of space, so we do it in a region of interest that contains all electric field sources). The Matlab tool requires boundary conditions for all surfaces, including the ones of the box. The potential at the plates was set to $\pm$1V, while the one at the box was specified in the following way

$$hV\big|_{Sbox} = r$$

by setting $h = 0$. One could think that this is the same as setting V directly to zero. However this is not true. That would be the equivalent of "making the box out of metal", which would produce reflection of the electric field, and the results would be totally different to the ones we hope to obtain. Boundary conditions must be set as if the box was not there at all.
Results of the 2D simulations with two plates are shown in FIG.1. A quick test was performed in order to check the validity of this solution. Gauss' law was applied integrating the field surrounding one of the plates and the capacity of the system was extracted. In two 2D this means

$$C = \frac{Q}{\Delta V} = \frac{\epsilon_0}{\Delta V}\oint \vec{E}\cdot\hat{n}dl$$

Then, this result was compared with the theoretical capacity that can be calculated through

$$C = \frac{\epsilon_0 S}{d}$$

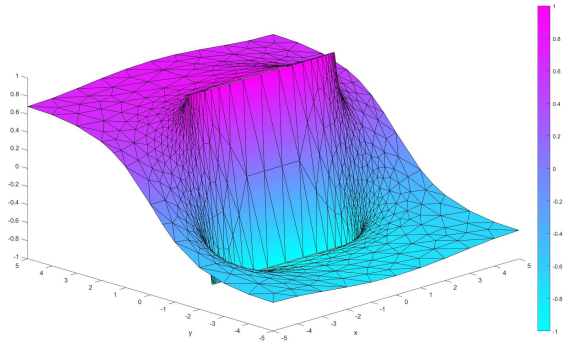


FIG. 1: Potential created by two 2D plates using FEM

[1] For more information see reference page for pdetoolbox at http://es.mathworks.com/products/pde/

where $S$ is the plate surface (longitude in 2D) and $d$ is the separation. Results matched quite well the expectations, taking into account that the theoretical derivation is also approximate (infinite-plate approximation).

### B. Method of moments

As said before, the goal is to find the potential function at every point in space $\Phi(\vec{r})$. From Poisson's equation we know that

$$\nabla^2\Phi(\vec{r}) = -\frac{\rho(\vec{r})}{\epsilon} \quad \Rightarrow \quad \Phi(\vec{r}) = \int_S \rho(\vec{r\prime})G(\vec{r}-\vec{r\prime})d\vec{r\prime}$$

where $\rho$ is the charge density and G is Green's function, which fulfils

$$\nabla^2 G(\vec{r}) = -\frac{\delta(\vec{r})}{\epsilon}$$

We have therefore translated our problem into finding the expression of $\rho$. There are some known boundary conditions that can be applied: the potential at the electrodes. This means that we know the solution of applying a certain functional $\mathcal{L}$ to $\rho$

$$\mathcal{L}\rho = \int_S \rho(\vec{r\prime})G(\vec{r}-\vec{r\prime})d\vec{r\prime}\Big|_S = \Phi_0$$

Conversion from a differential to an integral equation is very convenient since the meshing must only be applied over the electrode's surfaces, unlike before, when the mesh was made all over the volume.
The method of moments provides with an effective strategy to solve this kind of integral equation, by approximating the unknown function by a summation of basis functions, weighted by certain coefficients that will become our new unknowns.

$$\rho(\vec{r}) \approx \sum_{j=1}^{N} a_j f_j(\vec{r})$$

Substituting into the functional equation one can obtain

$$\Phi_0 = \mathcal{L}\rho \approx \sum_{j=1}^{N} a_j \mathcal{L} f_j = \tilde{\Phi}_0$$

The next step is to impose zero residue between the approximate and exact solution, using some weight functions $W_i$

$$R = \Phi_0 - \tilde{\Phi}_0$$

$$\langle W_i, R\rangle = 0 \qquad i = 1, .., N$$

where the scalar product has been defined as the internal Hermitian product

$$\langle f, g\rangle = \int f^*(\vec{r})g(\vec{r})d\vec{r}$$

Ultimately, this method will yield a linear system that can be reached by writing the matrix form of the equation

$$\langle W_i, \Phi_0\rangle = \langle W_i, \tilde{\Phi}_0\rangle = \sum_{j=1}^{N} a_j \langle W_i, \mathcal{L}f_j\rangle$$

$$A_{ij} = \langle W_i, \mathcal{L}f_j\rangle \quad b_i = \langle W_i, \Phi_0\rangle$$

So we obtain the simple relation

$$Aa = b \quad with \quad A \in \mathbb{M}_{N\times N} \quad a, b \in \mathbb{R}^{N\times 1}$$

A more detailed derivation of this can be found in [1]. Please note that Green's function is different depending on the dimensions of space. In 2D it has logarithmic form, while in 3D it has the well known $r^{-1}$ dependence

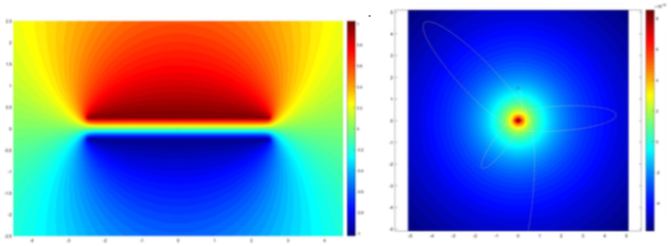
$$G_{2D}(\vec{r}) = -\frac{ln(r)}{2\pi\epsilon_0}$$

$$G_{3D}(\vec{r}) = -\frac{1}{2\pi\epsilon_0 r}$$

## II. SIMULATION RESULTS

Once the MoM algorithm was implemented, a number of tests were performed, with increasing level of difficulty (and hence resemblance to reality).
First, some 2D simulations were run in order to check that the MoM algorithm was correct. In this case, rectangular pulses were used as basis functions, while delta functions acted as weight functions (point matching). A two-plate geometry was created in order to compare the results to the one previously obtained with FEM. Next, a small test on the ion movement was performed, using a small one-plate geometry and observing the resulting trajectories. This two simulations are shown in FIG. 2. Please note that orbits in FIG. 2(b) are not as one would expect, given the logarithmic dependency of the potential with respect to the radial distance.

With regard to 3D simulations, plane geometries were used to start with. In this way it could be confirmed that the same results were obtained when plotting the



(a) Potential created by two 2D plates using MoM (b) Trajectory of an ion embedded in a one-plate potential

FIG. 2: MoM 2D simulations

potential at the XY plane only. The step from 2D to 3D may seem obvious, but a different kind of basis function needed to be used. Instead of the previous rectangular pulses, triangular pulses with constant height were employed. This meant computing the integral of a function proportional to r inside a triangle. Fortunately, the analytical deduction of this result was already derived by [2]. Following the results on this paper, a helpful Matlab function (already programmed by Professor E. Ubeda of the TSC department) was used.

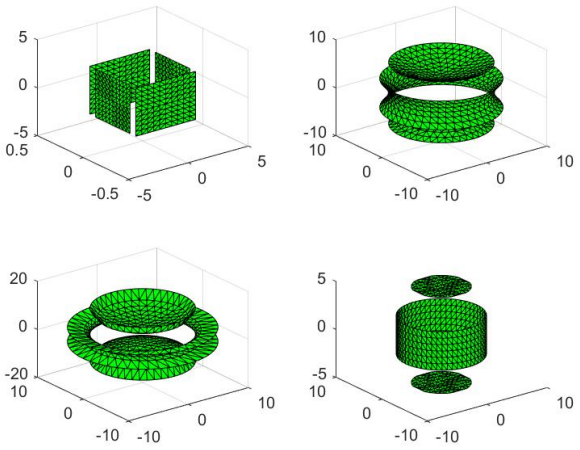


FIG. 3: Different geometries used

More complex geometries were created by a function that could be easily modified to produce hyperbolic, spherical or circular electrodes (see FIG.3). This function already provided the necessary data for the meshing. Special attention was given to the hyperbolic electrodes, which were used in further, more complex codes. However, all geometries were tested. In these cases, potential could be visualized by plotting it on the XZ plane (since there is revolution symmetry).
The aim of using these geometries is creating a saddle point. The quadrupole ion trap, changes the voltage applied to the electrodes in such a way that one can imagine a rotating saddle point with a sliding "ball" (the ion) on top of it, which moves dragged by gravity but never seems to fall, because the potential energy minima rotate periodically. In these first simulations potential was static, so this imaginary "ball" should fall "downhill", even with an initial velocity in the "uphill" direction. This was proven in a set of animations, whose result is shown in FIG.4

## III. ION TRAP DESIGN

Quadrupole ion traps often have hyperbolic geometries, since the resulting equations can be solved analytically. This kind of approach was not addressed in this project, as it was purely numerical. However, for a more realistic study, hyperbolic electrodes were used in the final and definitive simulations, so the results could be compered with the ones in [3].

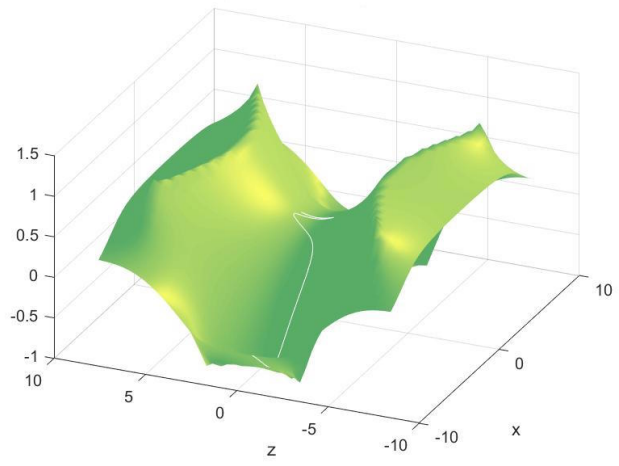



FIG. 4: Trajectory of an ion on a saddle point

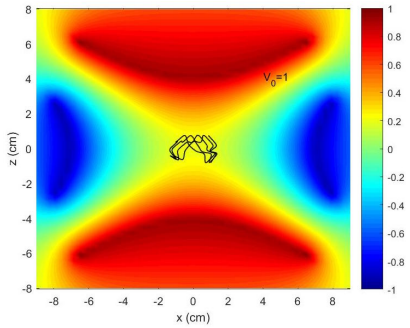
The difference in these codes with respect with the previously explained ones is that a time dependence of the potential applied to the electrodes was introduced. This time dependence was of the form

$$V(t) = V_0 \cos(\Omega t)$$

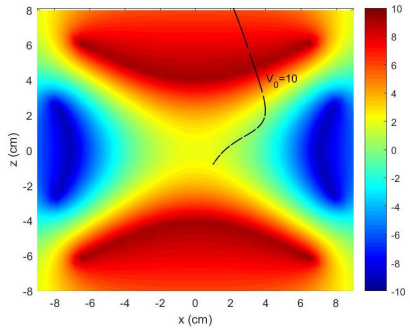
as used in [3]. Of course, this was translated into discrete time, calculating the potential at all points in space for a fixed number of voltage stages (let us call this number M) within one period. At this point, computational cost was a key element to take into account. Matrix $A$ of MoM was only computed once, and then the system was solved for all the M different independent coefficients vector $b$. The results were stored in the most efficient way possible. This is the equivalent of saying that we have given movement to the previously static saddle point in FIG.4.

It is important to notice that this static problem treatment that has been used up to this point is only valid given the low variation frequency of the voltage at the electrodes. In reality, information takes some time to travel from one place to another. By the time the change in the potential reaches the ion, a different voltage state could have already been applied to the electrodes. This means that there is a time delay (that can be regarded as a phase difference) between the harmonic variation of the potential at the electrodes and at the ion's position. The key here is to calculate how big is this delay compared to the oscillation period $T = \frac{2\pi}{\Omega}$. If we denote $l$ as the distance between the ion and the electrodes, one can say

$$\frac{\Delta t}{T} = \frac{l\Omega}{2\pi c} = \frac{l}{\lambda}$$




(a) Ion remains confined for $V_0 = 1V$




(b) Ion scapes for higher voltages, in this case $V_0 = 10V$

FIG. 5: Ion trajectory with different maximum voltages

Since the size of the trap is of the order of centimeters, and wavelength is 300m (1MHz frequency was used) we can completely neglect this time delay, i.e. we can pretend that the oscillations are in phase. If this were not the case, solving the equation as if the system were a succession of static stages would be incorrect.

The ion's trajectory was calculated taking small time steps. A key parameter was the number of time steps it took for the voltage stage to change from one to the next (remember that the M different stages were previously calculated). In any case, a linear interpolation from one stage to the next was implemented, in order to avoid abrupt changes in the potential.

These parameters were adjusted in order to fit the desired frequency of a voltage cycle, which is specified in [3]. What can be set is the maximum amplitude for the electrode's voltage ($V_0$); 1V showed to be enough to confine the ion. Higher voltage is not effective, since the acceleration we inflict on it is too high and the probability of it scaping increases (see FIG.5).
Having successfully simulated an efficient ion trap for one ion, the next step was to take into account the effect of several ions moving at once. The question was weather or not the repulsive force applied to each ion because of the others was strong enough to interfere with the trap's ability to confine them. Simulation results showed that for a small quantity of ions, the field created by each of them was weak enough to neglect it when compared with the one due to the electrodes, which means they will not affect the capacity to confine ions. This, of course is a result for a very low concentration of ions.

## IV. CONCLUSIONS

The goal of this project was to learn the basic numerical techniques that are commonly used in physics and engineering for electrostatic problem resolution. A scalar problem was chosen in order to avoid dealing with dispensable difficulties introduced by vectorial calculations. Starting from 2D, codes had to be optimized in order to reduce unnecessary computational cost, which would be unacceptable in 3D. Solving such a technically "simple" example gives an idea of all the efforts that must be made in order to compute more complex problems in computational electromagnetics, which have direct applications such as antenna analysis.
Final results gave the possibility to redesign some trap parameters, in order to improve its performance. Further work could be made in the last topic that was mentioned, regarding higher ion density.

[1] A. Cardama, L. Jofre, J.M. Rius, J. Romeu *Antenas*, Edicions UPC, Barcelona 1998
[2] S. Rao, A. Glisson, D. Wilton, B. Vidula *A simple Numerical Solution Procedure for Statics Problems Involving Aritrary-Shaped Surfaces* IEEE Transactions on Antennas and propagation, VOL. AP-27, NO.5, September 1979
[3] G.Werth, *Basics of Ion Traps*, Johannes Gutenberg University, Mainz