Grid-Free Treecode Method in Particle-In-Cell (PIC) Simulations

Janez Krek¹, Nikola Jelić¹, ², Jože Duhovnik¹

¹University of Ljubljana, Faculty of Mechanical Engineering, Aškerčeva 6, SI-1000 Ljubljana, Slovenia, janez.krek@lecad.fs.uni-lj.si, joze.duhovnik@lecad.fs.uni-lj.si

²Association EURATOM-ÖAW, Institute for Theoretical Physics, University of Innsbruck, A-6020 Innsbruck, Austria, nikola.jelic@uibk.ac.at

ABSTRACT

In this work we present basic principle of treecode (TC) method, its advantages and drawbacks in its applications for laboratory and fusion plasmas. Particular task done in this work is to create a small but efficient PIC simulation program in order to obtain more data than currently possible with other such programs available today, and to compare results of our program with results obtained with other methods and packages. The program is intended to be used in plasma-wall investigations of laboratory, technology oriented and fusion plasmas.

1 INTRODUCTION

In their historical article from 1986, Barnes and Hut [1] presented the treecode (TC) method as a method to reduce number of necessary calculations when describing interactions between bodies and/or particles in various systems. Basic idea of the method is to replace, whenever it is possible, particle-to-particle (P-to-P) interaction with particle-to-cluster (P-to-C) interaction. Conditions when clusters can be used instead of particles mainly depend on “size” (diameter) of a cluster and a distance between particle and cluster center point. In terms of accuracy and speed of the method, treecode method can be positioned between so called “direct summation” (known also as “direct integration”), with typical process of $O(N^2)$, and PIC method [2, 3, 4], with typical process of $O(N \log N)$. However, the main advantage of the TC method is its capability to deal with simulations that have big gradients in calculated values. Namely, while resolving short scale structures in plasmas, like sheaths and double layers, via simulation method, proper selection of a grid is of decisive importance. For resolving the sheath via numerical method with enough high accuracy, Kos et al. [5] used to employ a non-uniform grid where the length of grid cells, as approaching to the wall, was less than $10^{-7}$ of the total length of the system. In stationary state, obtaining the solution required several days long calculations per processor for a single case. In non-stationary simulations high resolution thus appears to be a much more demanding and expensive task, that we expect to carry out much more efficiently with tree code method.
1.1 Theoretical model

Our theoretical model is a plane diode. It works with mono-energetic electrons emitted from left-hand side of a diode. This system may be described via energy conservation and Poisson equation:

\[ n_0 u_0 = n u ; \quad \frac{m u^2}{2} - e\Phi = \frac{m u_0^2}{2} ; \quad \frac{d^2 \Phi}{dx^2} = \frac{e}{\varepsilon_0} n \]  

Equation (1)

Where equations (1) can be quickly transformed into:

\[ \frac{d^2 \Phi}{dx^2} = \frac{e}{\varepsilon_0} \frac{n_0}{1 + \frac{2e\Phi}{mu_0^2}} \]  

Equation (2)

It is, furthermore, convenient to introduce dimensionless variables:

\[ \eta = -\frac{2e\Phi}{mu_0^2} ; \quad \eta_m = -\frac{2e\Phi_{\text{min}}}{mu_0^2} ; \quad \lambda^2 = \frac{mu_0^2\varepsilon_0}{2n_0e^2} \]  

Equation (3)

\[ \xi = \frac{x}{\lambda} ; \quad \xi_m = \frac{x(\Phi_{\text{min}})}{\lambda} ; \quad \xi_a = \frac{x_a}{\lambda} \]  

Equation (4)

With dimensionless variables from equations (3) and (4), equation (2) becomes:

\[ \frac{d^2 \eta}{d\xi^2} = -\frac{1}{\sqrt{1 - \eta}} \]  

Equation (5)

with solution:

\[ \xi - \xi_m = \pm\frac{2}{3} \left( \sqrt{1 - \eta} + 2\sqrt{1 - \eta_m} \right) \left( \sqrt{1 - \eta} - \sqrt{1 - \eta_m} \right)^{(1/2)} \]  

Equation (6)

The solution of equation (6) is shown in Figure 1. It should be noted that a detailed analysis shows, that with increased density, the stationary solution is symmetric about the center of the diode and is obtained with maximum possible potential, \( \eta_m = 3/4 \), while above certain critical
density the potential profile jumps to \( \eta_m = 1 \). Further increase of the density does not lead to further potential increase (as obvious from physical reason, namely the maximum potential deep can not be greater than the initial electron energy), but the potential minimum moves to the left. Detailed inspection shows that the way the potential deep depends on the history of the beam density, i.e., shows a clear hysteresis behaviour. This is known to be related to instability, which, anyway, can not be modelled by the present stationary equations. This is one of the basic motivations to use this model for resolving non-stationary behaviour via simulation codes as follows in sections bellow.

### 1.2 Particle in cell (PIC) method

Particle in Cell (PIC) codes, found applications in a wide range of plasma investigations. They are based on computer particles, i.e., so-called “super-particles”, instead of “real” particles, as a solution to the problem of extremely huge number of real particles that requires too large computational resources. Thus a single “super-particle” contains as many as \( 10^6 \) “real” particles. Despite this method, in many applications, especially in fusion field of investigations, PIC simulations still take a long time to run. New parallel computing can reduce this time dramatically, however, the price of performing simulations is still very high. However, approximation involved with the concept of a super-particle remains intrinsically present in these codes, limiting inherently their accuracy. Another drawback is requirement for uniform grid in PIC codes, which limits user to one grid for whole simulation area. Anyway, for present purposes, we show our results of a short-cut diode in plane geometry obtained by a PIC code in Figure 2. The diode works with electrons emitted from left-hand side of mono-energetic electrons with initial velocity \( 4.2 \times 10^5 \text{m/s} \) (corresponding to \( 1 \text{eV} \) energy). As seen in Figure 2, during an initial stage of several \( \text{ns} \) the system is filled with particles, leading to increased space charge which produce a potential dip as shown in Figure 2. Once the space charge is high enough potential dip becomes sufficiently strong to prevent new coming electrons to overcome the potential barrier, i.e., to repel them back to the emitting electrode. This leads to movement of the potential barrier and consecutive decrease of density and the strength of potential barrier, leading again to new condition for deeper penetration of electrons into the diode. As a result, a periodic oscillation of the entire system is established, as expected from the theoretical analysis. In addition, Figure 2 is a reference example in context of results bellow.

![Figure 2: Number of “super-particles” in diode in relation to time (a) and potential profiles (b) in various simulation steps.](image-url)
1.3 Treecode method

Basic idea behind treecode (TC) method is to replace particle-to-particle interaction with particle-to-cluster interactions in such a way, that we reduce the number of necessary calculations required without making too much error. With reduced number of clusters in comparison to number of particles, number of required interactions (and thus required simulation time) is reduced. Number of iterations for calculating interaction between particles is in range of $O(N^2)$, with the use of TC method, number of interactions particle-to-cluster falls in range of $O(N \log N)$ - this is a huge reduction in number of required calculations. In current plasma simulations, where number of particles is of the order of $10^{22}$ (or $10^{17}$ super particles), number of interactions particle-to-particle is large even for today’s super computers, or at least computers that are usually used to run simulations. TC method is scalable and can run on parallel computers computers \[6\]. TC method is most simple in 1-D space, but can also be extended in a way, that can be used in 2-D or 3-D space \[7\]. The difficult part in extension into more dimensions is not TC method itself, but underlying equations that are solved using treecode method. Due to the nature of TC method, TC method gives precisely the same results in 1-D space as direct summation method \[8\]. Method introduces another level of approximation to PIC simulation, in addition to already widely used approximation: using computer particles instead of “real” particles. With smart selection of parameters for treecode method, we expect the impact of TC approximation to simulation results to be minimum. TC method is grid-less \[9\]; for running simulation one does not need to define a grid for calculating values of interest. Main computation of particle interactions (positions, velocities) are done using generated tree and without any grid. Grid is needed just as an auxiliary tool in post-processing.

![Figure 3: Example of tree representation (a) of square domain (b) (image taken from \[10\)](image)

1.3.1 Building the tree

Main tool in TC method is a tree, which is a hierarchical representation of particle positions in simulation system (example of 2-D system is shown in Figure 3b). Because tree depends on particle positions, it is necessary to build the tree in each simulation step. Tree is composed of nodes, links between nodes and leafs (Figure 3h), with each node can have 0 or more children. Maximum number of children in each node of a tree is defined by dimensionality of domain, for which tree is created, and can be up to: 2 children for 1-D domain (also called binary trees), 4 children for 2-D domain (also called quad-trees) and 8 children for 3-D domain (also called octrees).

There are nodes in a tree that have special names: nodes without children are called leafs (nodes named $a$, $h$, $i$, etc. on Figure 3) and node without a parent is called root node (node
Figure 4: Generation of tree and dividing domain into sub-domains. Nodes labeled as “0” are called “root” nodes, nodes without sub-nodes are called “leafs”.

named root on Figure 3. The tree can be asymmetrical and does not need to be full - there can be nodes in tree that have less children than maximum number (for example: 1 child instead of maximum 4 children for 2-D system) - see Figure 4.

Building of a tree in 2-D space begins with dividing a simulation area to 4 equal sub-areas and particles inside sub-areas are assigned to each sub-area. Then each sub-area is checked if it should be subdivided further. This process is repeated until there is no area to be subdivided further (see Figure 4: dividing area from level 2 to level 3). First main TC parameter defines a test, if certain area should be subdivided further or not. With this parameter, one can influence on number of tree nodes and how large will be final tree. Larger tree results in slower simulation time and less gain from treecode method. Smaller tree results in faster simulation times and more approximation because using particle-to-cluster interaction instead of particle-to-particle interaction.

2 RESULTS

To test TC method in simulations, we developed our program from scratch in program language C, compiled on Linux, with all advantages of TC method. Program enables user to define main simulation parameters in input file: densities, simulation system sizes, output variables,
parameters for tree generation, etc. In preliminary tests, it showed some advantages in comparison with available free PIC and treecode packages codes. Program enables computation of various plasma parameters: particle positions, densities, velocity distribution functions (VDFs), temperatures, fluxes, etc. with a high resolution in regions of strong plasma and field gradients. Results obtained from present 1-D version are precisely the same as results from (very demanding) direct summation method. This feature arises from the fact that in 1-D geometry, underlying numerical method does not have additional errors [8]. Our test case was 1-D vir-

Figure 6: Potential (a1, a2), phase-space (b1, b2) and particle density (c1, c2) in one cycle. Left column when potential is increasing and right column when decreasing.

Figure 6: Potential (a1, a2), phase-space (b1, b2) and particle density (c1, c2) in one cycle. Left column when potential is increasing and right column when decreasing.

Figure 6: Potential (a1, a2), phase-space (b1, b2) and particle density (c1, c2) in one cycle. Left column when potential is increasing and right column when decreasing.
state. We selected $t = 0.45\, \text{ns}$ of simulation time. Figure 5 (MatLab code - 5a; C program - 5b) show phase-space diagram for selected time from start of simulation. Once a quasi-stationary state is established, the system oscillates with a period of $T = 3.5\, \text{ns}$. When simulating virtual cathode, system comes to quasi-stationary state and then oscillates. In Figure 6 we show values for phase-space diagram, potential profile and density in various times during oscillation cycle. Values for density are normalized to total number of particle in system and density peak is cut-off due to readability reason.

In the left column we show the potential profiles, densities and phase-space diagrams during the first stage of the cycle (increasing potential deep from minimum to maximum value, i.e., during the decreasing number of particles in system). In the second column we show the same quantities during the opposite stage (decreasing potential, i.e., increasing number of particles). It is a little tricky to interpret simultaneous behaviour of potential profile, density and phase-space in time on static figures, since the positions of particles are not simply related to the instant potential profile, but are a consequence of the whole history during the period of cycle. That is why one has to obtain and store a huge number of data for making appropriate time dependent records for better observing qualitative features of such time-delayed effects in any point of observation of said quantities.

3 DISCUSSION

Nowadays, simulations require more and more computer power and there are requirements to perform more and more accurate simulations. There are needs to try and test methods from other science fields in a field of plasma simulation. We tried and tested treecode method using our new program written in C to investigate a particular physical scenario of short-cut plane diode. This apparently simple system is in fact rich in various phenomena which might arise due to e.g., external voltage, initial particle velocities and temperatures, presence of various kind of particles, etc. In fact this is a basis for any further investigation of plasma sheath properties which are of extreme importance for bounded plasmas such as produced in laboratory, industrial and fusion devices. We have shown the most simple, yet still complex case of a virtual cathode formation appearing in a short-cut diode as the space charge effects become considerable in diode. We have shown our analytic theoretical solution of non-stationary behaviour taking role providing a critical value of the space charge effects is reached with increased electron injection. Since non-stationary behaviour details can not be resolved via analytic method we employ a standard PIC method for this purpose. Furthermore, we use the treecode method as obtained by [8] to confirm these results, as well as our simulation program for obtaining extended results, i.e., potential profiles, densities and other averaged quantities that might be derived from simulation, like velocity distribution functions and their moments. Due to these mentioned features of our code we found it superior in comparison to other methods. In this stage of development our program still does not have the foreseen functionality included. On one hand, some of them which are included still have to be tested and benchmarked, i.e., possibility of using more species to use large number of particles (higher than $10^{20}$ particles), computation of velocity distribution functions (VDFs), temperatures, fluxes, etc. On the other hand, some features are still to be done in the nearest future: to add the external magnetic field, versatile source distributions, particle-boundary interaction (reflection, absorption), field-boundary interaction (electron fields emission), Coulomb collisions, MCC (Monte-Carlo collisions), etc. A special feature planned in our work is a possible coupling with currently developing hybrid code OOPD1 of PTSG group from USA.
ACKNOWLEDGMENTS

This work was supported by the European Commission under the Contract of Association between EURATOM and the Austrian Academy of Sciences. It was carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work was also supported by the Austrian Science Fund (FWF) under project P19333-N16.

REFERENCES


