SANDIA REPORT

SAND2011-2969
Unlimited Release
Printed April 2011

Numerical Experiments on Unstructured PIC Stability

David M. Day

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-2969
Unlimited Release
Printed April 2011

Numerical Experiments on Unstructured PIC Stability

David M. Day
Applied Math and Applications Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-9999
dmday@sandia.gov

Abstract

Particle-In-Cell (PIC) is a method for plasmas simulation. Particles are pushed with Verlet time integration. Fields are modeled using finite differences on a tensor product mesh (cells). The Unstructured PIC methods studied here use instead finite element discretizations on unstructured (simplicial) meshes. PIC is constrained by stability limits (upper bounds) on mesh and time step sizes. Numerical evidence (2D) and analysis will be presented showing that similar bounds constrain unstructured PIC.

Acknowledgment

I wish to thank Matt Hopkins (1516) for suggesting these experiments and for his continuous and constructive feedback. as well as Tom Hughes (2735), Jeremy Boerner (1516), Russel Hooper (1445) and Paul Crozier (1426) for conversations that significantly improved this document.

Contents

Nomenclature . . . 7
Introduction . . . 9
Description of experiments . . . 10
Meshes Used in the Study . . . 12
Primary Results . . . 14
Heating Rates . . . 35
Conclusions . . . 43

Appendix

A Physical Constants . . . 48
B Conservation Laws . . . 49

Figures

1 Instantaneous electron temperature, very coarse mesh . . . 15
2 Instantaneous ion temperature, very coarse mesh . . . 16
3 Instantaneous and averaged electron temperature . . . 17
4 Structured and unstructured PIC: electrons, very coarse mesh . . . 18
5 Structured and unstructured PIC: ions, very coarse mesh . . . 19
6 Instantaneous electron temperature, coarse mesh . . . 20
7 Instantaneous ion temperature, coarse mesh . . . 21
8 Instantaneous and averaged electron temperature, coarse mesh . . . 22
9 Structured and unstructured PIC: electrons, coarse mesh . . . 23
10 Structured and unstructured PIC: ions, coarse mesh . . . 24
11 Structured and unstructured PIC: electrons, neutral meshes . . . 25
12 Structured and unstructured PIC: ions, neutral mesh . . . 26
13 PIC: electrons, fine meshes, variable domain . . . 27
14 Structured and unstructured PIC: ions, fine mesh . . . 28
15 Structured and unstructured PIC: electrons, very fine meshes . . . 29
16 Structured and unstructured PIC: ions, very fine mesh . . . 30
17 Electron temperature versus weight, fine mesh . . . 31
18 Ion temperature versus weight, fine mesh . . . 32
19 Electron temperature versus weight, very fine mesh . . . 33
20 Ion temperature versus weight, very fine mesh . . . 34
21 Electron temperature versus temperature, very coarse mesh . . . 36
22 Electron temperature versus temperature, very coarse mesh . . . 37
23 Electron temperature versus initial temperature, mesh . . . 38
24 Heating Rate versus Debye Length . . . 39
25 Heating Rate versus Courant number . . . 40
26 Heating Rate versus Cost per Second . . . 41

Tables

1 Mesh Topology 12
2 Mesh Element Areas 12
3 Mesh Edge Lengths 13
4 Growth Rate 42
5 Raw Run Times 44
6 Raw Solve Times 44
7 ScaledMesh: num Particles 45

Nomenclature

temperature 1 electron volt = 11605 Kelvin

electron density 10^{10}cm^{-3}

Debye length $\lambda_D = .74339 \times 10^{-2}$ cm

plasma frequency $\omega_p = 5.6415 \times 10^3/\mu\text{sec}$

thermal velocity $v_{Th} = \lambda_D \omega_p = 41.938$ cm/μsec

Introduction

The equations governing plasmas and standard Particle-Mesh [1, 5] numerical modeling methods are reviewed. Particle Mesh methods are sometimes called Particle in Cell (PIC) methods, particularly for rectangular domains and tensor product meshes. The simulation methods discussed here involve a finite number of numerical particles at locations $\{x^i\}$ with velocities $\{v^i\}$. The plasmas considered here have ions (protons) of charge e and electrons of charge $-e$. The particle weight is the number of particles per numerical particle.

The charge density field ρ averaged over each finite element K satisfies

$$\int_K \rho(x) d^3x = \sum_{x^i \in K} q^i. \tag{1}$$

Gauss' Law relates the electrostatic potential ϕ and the charge density ρ,

$$\mathrm{div}\varepsilon E = \rho, \quad E = -\nabla\phi. \tag{2}$$

And Coulomb's force drives the particles,

$$m^i \frac{d}{dt} v^i = q^i E(x^i). \tag{3}$$

Experiments on the stability limits for unstructured particle in cell (UPIC) plasma simulations are reported. A plasma in steady state is modeled. In these simulations collisions are ignored. Instabilities may lead to spurious heating. The experiments on quasi-uniform meshes indicate that the stability limits for unstructured meshes and structured unstructured meshes are similar.

The time integration is implemented using the Störmler-Verlet method. PIC uses a finite volume discretization [1]. The analogous discretization on an unstructured mesh uses the Delaunay-Voronoi grids [10, 3]. The dependence of the stability limits on the mesh are not well understood.

Ordinary finite elements [11] are a simpler starting point. In the UPIC method here also uses a finite element discretization on a simplicial mesh. Particle density is constant within each element.

The numerical experiments described here all start from a model of a plasma in a square domain. A tensor product mesh is a structured mesh. The structured mesh of interest here is the triangle mesh derived from a tensor product quad mesh. Our domain is the unit square $[0,1]^2$, unless stated otherwise. Particles reflect off the walls (specular boundary conditions). The potential field vanishes at the boundary.

Each numerical particle represents many physical particles. The initial particle distribution is approximately Maxwellian. The particle trajectories are stochastic. Results are presented here in terms of the time evolution of the spatially averaged temperature. Though non-deterministic, the qualitative properties of the spatially averaged temperature trajectories are consistent across multiple runs.

If the time step size or mesh size is too large, T_e may increase spuriously. The Debye length (see A) is given by $\lambda = \sqrt{\varepsilon kT/(q^2 N_o)}$, or in words

$$\text{debyeLength}^2 = \frac{\text{permittivity boltzmann temperature}}{\text{numberDensity elementaryCharge}^2}.$$

If we start out on a mesh with elements of diameter h, $h > \lambda$, the simulation becomes unstable. This equation suggests that plasma will heat up by a factor of

$$(h/\lambda)^2, \tag{4}$$

and then reach a steady state.

The standard PIC methods are stable if certain stability constraints are satisfied including element diameters $h \leq \lambda_D$, time step sizes $dt \leq 2/\omega_p$ and statistical constraints on the particle weight, The plasma frequency in units of interest is determined in the Appendix. Our experiments indicate that list does not include a crucial constraint.

PIC is connected to explicit methods for hyperbolic equations through particle wave speeds. The number cdt/dx is called the Courant number. A Courant-Friedrichs-Lewy (CFL) condition is a necessary condition of the form $cdt \leq Cdx$ for the numerical stability of explicit time integration methods for hyperbolic equations. The CFL condition, $cdt \leq dx$ is necessary for the stability of the staggered leap-frog method (i.e. Verlet) applied to the second-order wave equation, $u_{tt} = c^2 u_{xx}$.

Ideally particle wave speeds c are connected to v_{Th} through the Maxwellian probability distribution,

$$\chi(c) = \frac{4}{\sqrt{\pi}} \frac{c^2}{2v_{Th}^2} e^{-c^2/(2v_{Th}^2)} \frac{1}{\sqrt{2}v_{Th}}.$$

Set $z = c/(\sqrt{2}v_{Th})$ and integrate by parts to find the cumulative distribution

$$\int_0^{a\sqrt{2}v_{Th}} \chi(c)dc = \frac{4}{\sqrt{\pi}} \int_0^a z^2 e^{-z^2} dz = \text{erf}(a) - \frac{2}{\sqrt{\pi}} a e^{-a^2}.$$

It's not known to the author how to choose a in general. However a so that $\int_0^{a\sqrt{2}v_{Th}} \chi(c)dc = 9/10$ is used in certain applications. In this way leap-frog is stable for 90 percent of the particles if the time step satisfies if

$$Cv_{Th}dt \leq dx, \quad C = a\sqrt{2}. \tag{5}$$

This corresponds to $a \approx 1.768$, and $C \approx 2.500$. The precise value of C is not known in general. The important point is that as long as $dx \leq 2C\lambda$, the CFL condition more severely constrains the time step size than the plasma frequency bound, $dx/(C\omega\lambda) \leq 2/\omega$.

Description of experiments

A set of experiments are reported that characterize the regions of instability for unstructured PIC. Issues related to using computational resources efficiently are also mentioned.

1. For fixed initial spatially averaged temperature ($T_e = 1$ eV), electron density (10^{10}), particle weight (10^5), study mesh sizes near λ_D, and time step sizes near $2/\omega_p$.

2. If T_e increases, a time step that was initially PIC stable (e.g. $\omega_p dt < 2$) may become unstable. The fixed time step may be stable at the beginning, but unstable at the end. However, we want to run one (or a few) simulations that use a (much) smaller time step so that is PIC stable throughout the run.

3. The mesh sizes used are approximately $dx/\lambda = \{1/16, 1/4, 1, 4, 16\}$, $dx = dx_{\max}$.

   The simulations involve different meshes of the same domain, typically a square centimeter. However simulations on the finer meshes often use a fixed approximately structured mesh of the domain $[0, n\lambda]^2$, where $n \in \{32, 64, 128\}$. The advantages of reducing the domain include more efficient use of computational resources and simplification of the meshing problem. A disadvantage of reducing the domain size is that it is necessary to demonstrate the extent to which the temperature trajectory is independent of the domain size.

4. With explicit time integration the time step size is limited, but the precise form of the bound is unknown. Results are reported for time step sizes $dt = \{1/100, 1/10, 1, 10, 100\}2/\omega_p$.

   The spatially averaged temperature is compared to the temperature averaged both in space and over a fixed number of time steps (100). Some unstable simulations terminate early. Most simulations continue until the temperature increases above the theoretically maximum heating, equation (4).

5. For the initial run-through, use a particle weighting of 10^5. If economical, compare particle weights of $10^{\{3,4,5,6,7\}}$ to confirm that particle weighting sensitivity disappears for $dx < \lambda$. With large weights, e.g. 10^7, heating is expected.

The initial plan was to use the element temperatures to determine T_e at a given time. However, numerical evidence indicates that, with fixed weight, as the mesh size decreases, the number of numerical particles per element decreases, and it becomes impossible to accurately determine an element temperature. Here, element temperatures are used on the coarse meshes $h/\lambda = 4, 16$ only. On the other meshes, T_e is determined directly from the particles. Element temperatures are reliable when there are 33 or more particles per element, on average. On the other hand, element temperatures are unreliable when there are 1.5 or less particles per element, on average. Note also that for large scale simulations writing all the particles to one file is not scalable.

The sensitivity of the results to the initial particle distribution. has not been quantified. Some of the simulations reported here have been performed with different initial conditions, and in these experiments the spatially averaged temperature was insensitive to the particle distribution.

Meshes Used in the Study

In this section the structured and unstructured meshes of the unit square $[0,1]^2$ used in the simulations are described in detail.

The maximum edge length, h, is typically not much longer than the mean edge length. Maximum edge length is standard from the finite element theory for quasi-uniform meshes, but here it is a conservative measure. The mean edge and the standard deviation in edge length are also reported.

Table 1. The topology and diameter of the three coarsest pairs of structured and unstructured meshes are tablated.

Name	Structured?	h	number elements	number vertices	number edges
squareL16	Yes	.14	200	121	320
squareL16U	No	.14	150	92	241
squareL4	Yes	.038	2738	1444	4181
squareL4U	No	.038	2668	1403	4070
squareL	Yes	.0074	72200	36481	108680
squareLU	No	.0073	88186	44486	132671
squareLm4	Yes	1.86e-3	1158242	580644	1738885
squareLm4U	No	1.86e-3	1566338	784860	2351197

The number of elements in the structures meshes, $K = 2n^2$, for $n \in \{10, 37, 190, 761\}$. A mesh for the unit square with $h = \lambda/16$ would have $2(3044)^2$ elements.

Table 2. Element areas for the structured mesh are more uniform than for the unstructured mesh.

Name	Structured?	h	min	mean	max	std
squareL16	Yes	.14	4.9925e-03	5.0000e-03	5.0068e-03	1.9755e-06
squareL16U	No	.14	4.5585e-03	6.6667e-03	8.2574e-03	9.5309e-04
squareL4	Yes	.038	3.6474e-04	3.6523e-04	3.6562e-04	4.3852e-08
squareL4U	No	.038	2.5199e-04	3.7481e-04	5.5664e-04	2.3681e-05
squareL	Yes	.0074	1.3832e-05	1.3850e-05	1.3865e-05	3.2378e-10
squareLU	No	.0073	6.5645e-06	1.1340e-05	2.1014e-05	5.4564e-07
squareLm4	Yes	1.86e-3	8.6221e-07	8.6338e-07	8.6429e-07	5.0391e-12
squareLm4U	No	1.86e-3	2.4586e-07	6.3952e-07	1.2365e-06	7.8002e-08

`std` is Matlab's built in function for standard deviation. The documentation mentions some

normalization which I have not yet figured out, so I'm not stating the definition here.

Edges for the structured mesh have a bimodal distribution, which makes the structured mesh appear less uniform than for the unstructured mesh. in Table 3.

Table 3. Element edge lengths for the first six meshes.

Name	Structured?	min	mean	max	std
squareL16	Yes	9.9913e-02	1.1294e-01	1.4147e-01	1.9219e-02
squareL16U	No	9.1886e-02	1.2445e-01	1.4379e-01	1.0978e-02
squareL4	Yes	2.7003e-02	3.0692e-02	3.8237e-02	5.2538e-03
squareL4U	No	2.1613e-02	2.9446e-02	3.8418e-02	1.2482e-03
squareL	Yes	5.2584e-03	5.9873e-03	7.4462e-03	1.0268e-03
squareLU	No	3.4207e-03	5.1179e-03	7.3207e-03	1.4342e-04
squareLm4	Yes	1.3129e-03	1.4953e-03	1.8591e-03	2.5653e-04
squareLm4U	No	6.4303e-04	1.2153e-03	1.8599e-03	8.5331e-05

Simulations on a mesh of $[0,1]^2$ with $h = \lambda_D/16$ were deemed unnecessary, for the following reason, and were not performed. The reason is that simulations of equal fidelity are possible on a smaller domain. In the figure captions, "Dom$n$" refers to a mesh that is scaled down to $[0, n\lambda_D]^2$ where typical values are $n = 2^{5,6,7}$. In Figures 11 and 12, "Dom190" refers to the mesh squareL. In Figures 13 and 14 "Dom538" refers to the mesh squareLm4. The meshes "Dom190" and "Dom538" are in fact not scaled down. Results are presented on these meshes for comparison with smaller domains.

The meshes are quasi-uniform. An important issue not addressed here is the dependence on mesh grading of quantities such as the Courant Friedrichs Lewy condition.

Primary Results

In this section the results of the numerical experiments outlined in section are presented. The meshes used are described in section . Structured and unstructured meshes were compared in a mesh resolution study, Each figure depicts the average temperature as a function of time. Temperature is averaged over the area from the element temperatures. and sometimes in time as well. For finer meshes, temperature is computed from the particles directly. Time is measured in units of $2/\omega$. Note that $2/\omega = .35 \times 10^{-3}\mu$ sec.

The number of time steps was determined by trial and error. Total simulation times on the order of one microsecond are expected. It is meaningless to continue a catastrophically unstable simulation. But on the other hand, on coarse meshes it is important to determine the asymptotic nature of the heating. When reading the figures, it's important to bear in mind that the output frequency is usually once per 1000 time steps. Many different time step sizes are compared in one figure. Many figures use logarithmic time scales, which is unusual.

The spatially averaged temperature of the electrons is shown in electron volts. The initial electron temperature is always 1 electron volt. Initially the ions are frozen. The spatially averaged temperature of the ions is shown in Kelvins.

The first five figures show unstable simulations on a very coarse mesh, comparing different ways to average particle temperatures, as well as comparing structured and unstructured simulations. The next five figures concern a coarse mesh. These figures show three elemental quantities: spatially averaged electron temperature `T(em)`, electron temperature averaged in space and over a window of 100 time steps, `T(em)_100`, and spatially averaged ion temperature, `T(Hp)`. The last three figures show simulations on a (barely) stable mesh.

The $2/\omega$ time step stability limit is confirmed (approximately). The computations using unstable time steps heat catastrophically. Results are reported here for time step size 10 and 2000 iterations only on the coarsest mesh. The step size is much too large. For the other more computationally demanding problems, the simulation is halted once they become unstable. For the fine meshes, simulations are smaller time step sizes are run first until significant heating is observed. For the Debye stable meshes, the time steps are $[.01, .1]2/\omega$.

A fundamental premise of these experiments is to hold the particle weight constant and refine the mesh. As the number of particles per element decreases, the element temperatures become meaningless. On the stable meshes temperature is determined directly from the particles. No temporal averaging is done.

During the experiments, we made some major simplifications after completing the experiments at $16, 4$ and most of 1. We stopped looking at several parameters: unstable time steps, time averaging, and "structured" versus "unstructured" meshes. For the finer meshes, we concentrated on stable or nearly stable time steps, no time averaging, and only "structured" simplicial meshes.

The last four figures study the influence of the particle weight on the simulation. Unstable simulations are sensitive to particle weight, with lower weights generally reducing the severity

of the instability. On the other hand, the figures presented here show how increasing the particle weight can destabilize a stable simulation.

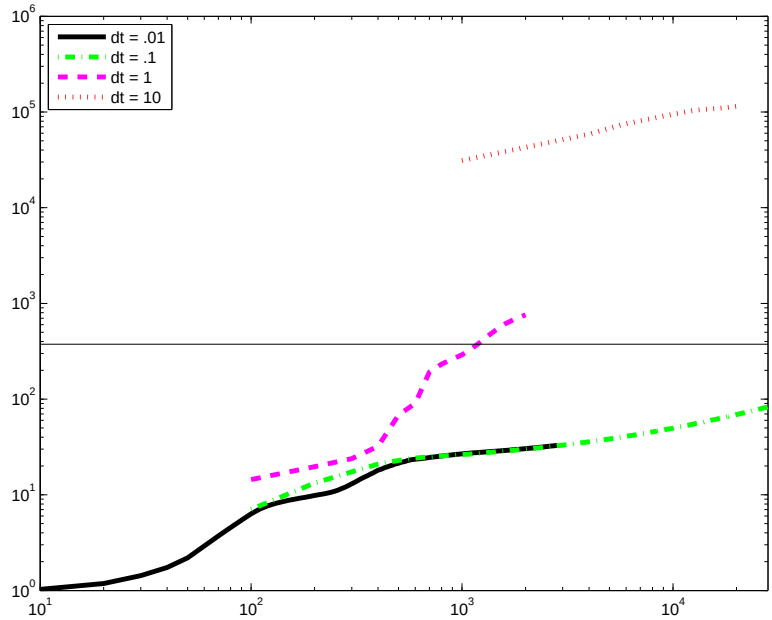

Figure 1. Spatially averaged electron temperature is shown at a function of time. The unstructured mesh contains elements of diameter at most $16\lambda_D$. The horizontal line marks the predicted level of heating. Time is measured in units of $2/\omega_P$. The simulations with time step size $dt = .01$ and $dt = .1$ are run for 282000 time steps. Consistent numerical heating is observed with $dt = .01$ (solid black) and $dt = .1$ (dash-dot green). However the simulations at $dt = 1$ (dashed magenta) and $dt = 10$ (dots red) heat up severely.

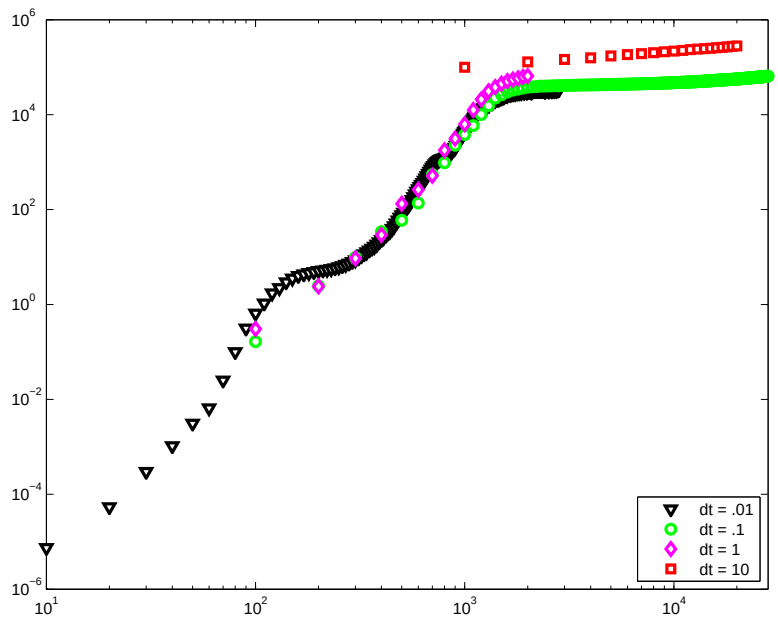

Figure 2. Instantaneous spatially averaged *ion* temperature (in Kelvins) is shown for the unstructured mesh with element diameters of at most $16\lambda_D$. Time step sizes are measured in units of $2/\omega_P$. From the left to the right, the triangles, circles, diamonds and squares correspond to time step sizes of $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$ respectively. The simulations at $dt = .01$ and $dt = .1$ are run for 282000 time steps.

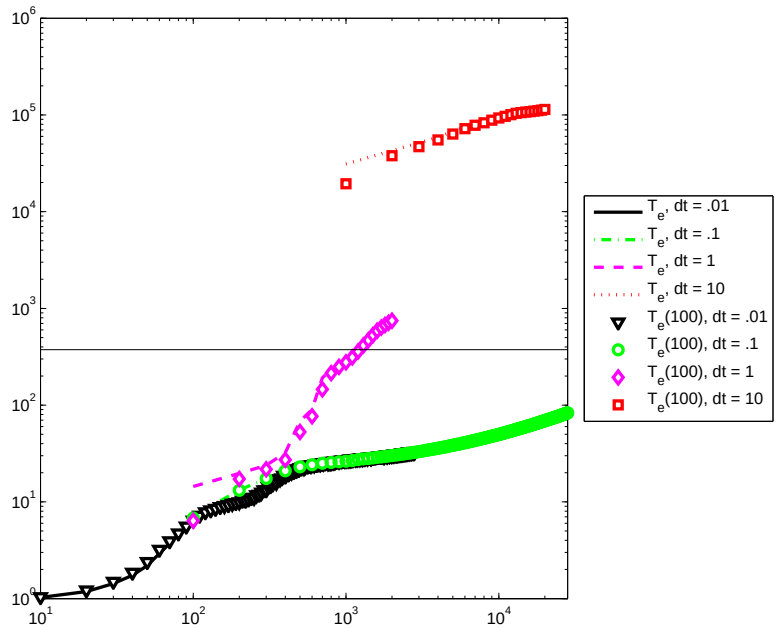

Figure 3. Temporal averaging slightly decreases the spatially averaged temperature. Here the mesh is unstructured with mesh size $h \leq 16\lambda$. The spatially averaged electron temperature, `T(em)` `(line plots) is compared to the spatial average of the temporally averaged (`100 `time steps) electron temperature T(em)_100 (symbols).`

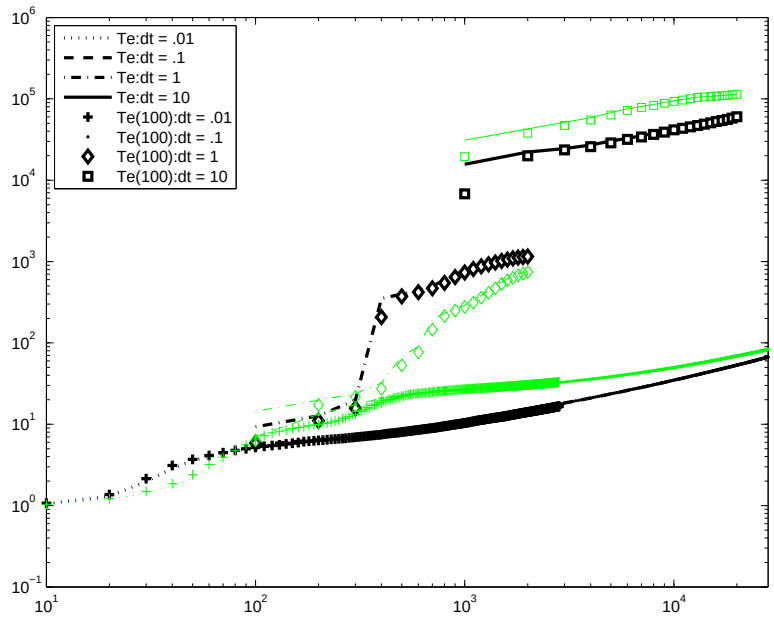

Figure 4. Similar heating is observed with an unstructured simplicial mesh and a structured simplicial mesh with the same maximum element diameter (here $h = 16\lambda$). Spatially averaged electron temperatures (lines) for the *structured* mesh are shown in black, with symbols used for the temperature averaged in both space and time. The similar results for an unstructured mesh are shown in green. Both the instantaneous and the time averaged electron temperatures are shown. At $dt = .1$ the heating is slightly worse on the unstructured mesh. In general qualitatively similar results are obtained with structured and unstructured meshes.

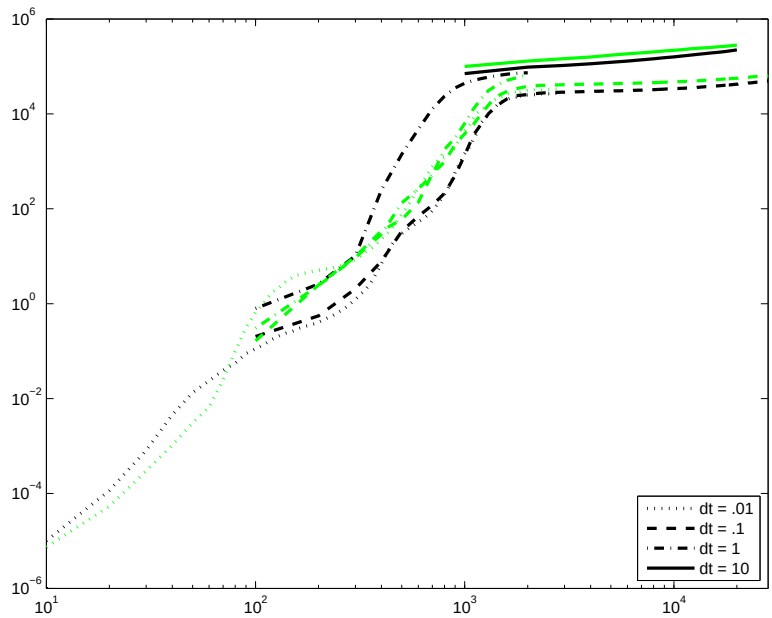

Figure 5. The heating of the spatially averaged ion temperature is compared for structured (black) and unstructured (green) meshes of element diameter $h = 16\lambda$. Initially the ions are cold. There is no temporal averaging.

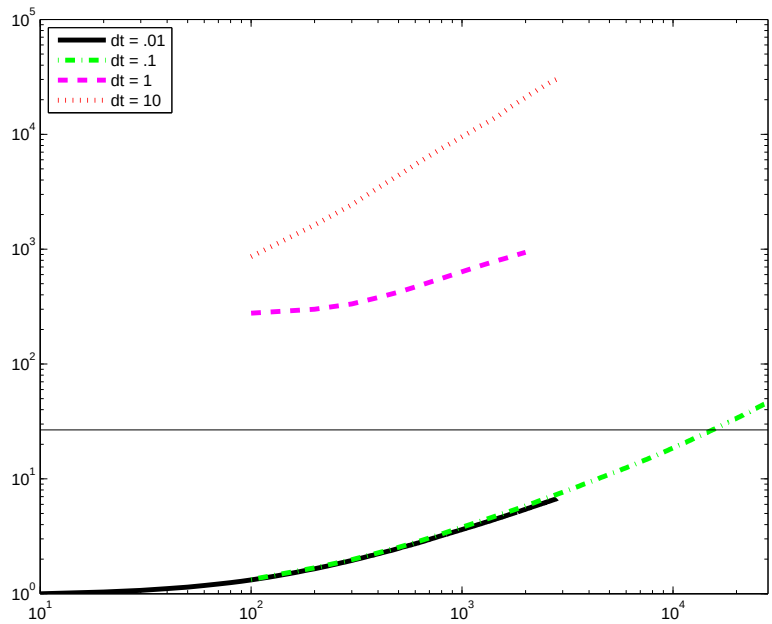

Figure 6. For the unstructured with mesh with $h \leq 4\lambda$ the spatially averaged electron temperature is shown The time step sizes are $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$. The corresponding numbers of time steps are $\{282000, 282000, 2000, 280\}$. The simulations at each time step are, or appear to be, heating past the predicted heating threshold.

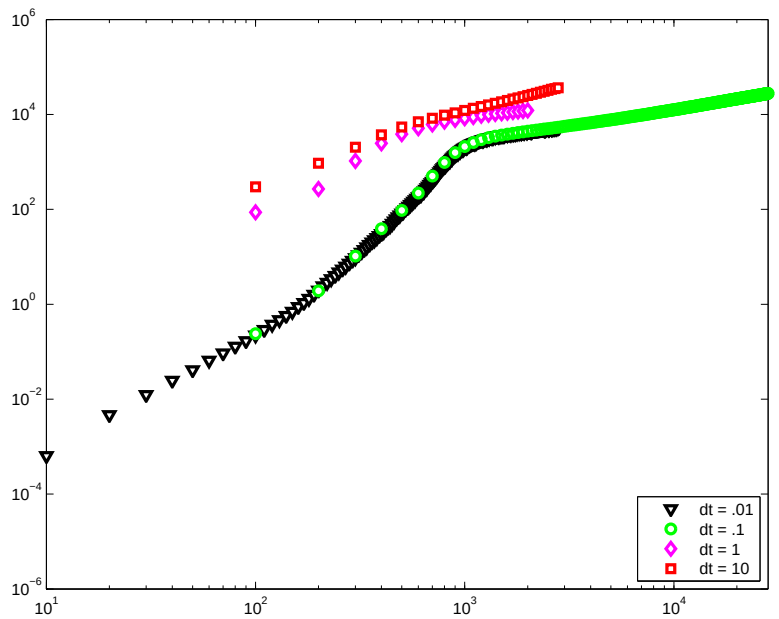

Figure 7. For the unstructured mesh with $h = 4\lambda$, the spatially averaged *ion* temperature is shown versus time. Note the heating at $dt = 1$ (diamonds) compared to $dt = 1/10$ (circles).

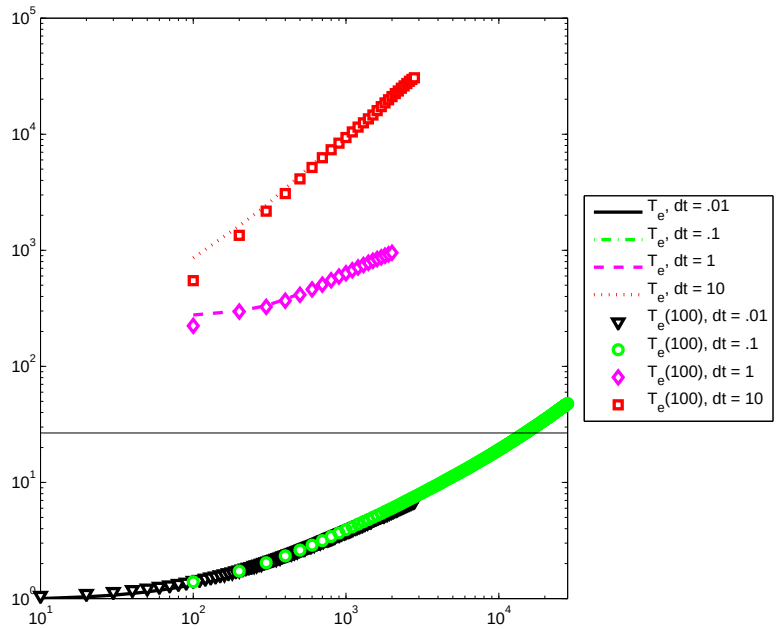

Figure 8. For the unstructured mesh with $h \leq 4\lambda$ the spatially averaged electron temperature is compared to the electron temperature averaged both spatially and over 100 time steps. The dotted, dashed, dash-dot and solid lines show the temperature averaged in space determined using time step sizes $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$ respectively. The triangles, circles, diamonds and squares show the temperature averaged in space and time determined using time step sizes $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$ respectively.

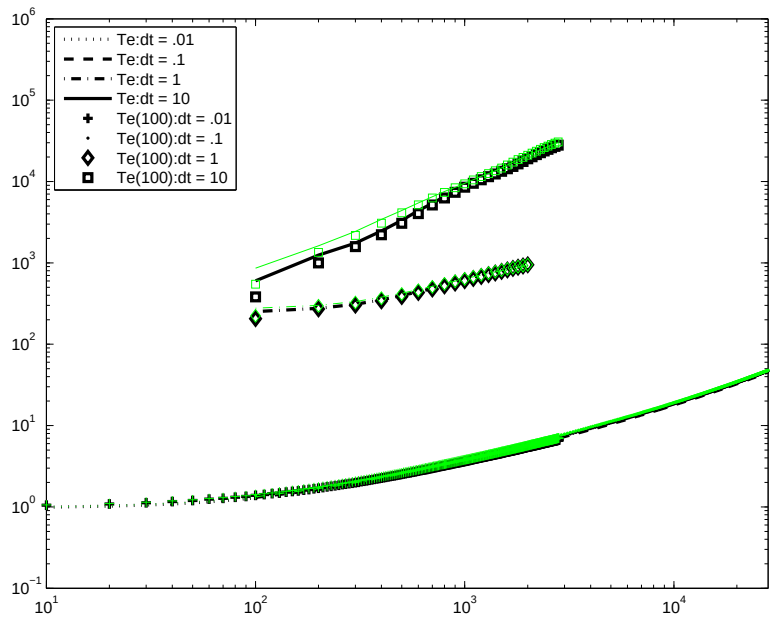

Figure 9. Electron heating is insensitive to quasi-uniform unstructured meshes and temporal averaging. Here simulations on a structured (black) and unstructured (green) mesh both with $h \leq 4\lambda$ are compared. Electron temperature is shown using time step sizes $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$, and averaged either in space (solid lines) or both in space and time (triangle,circle, diamond, square).

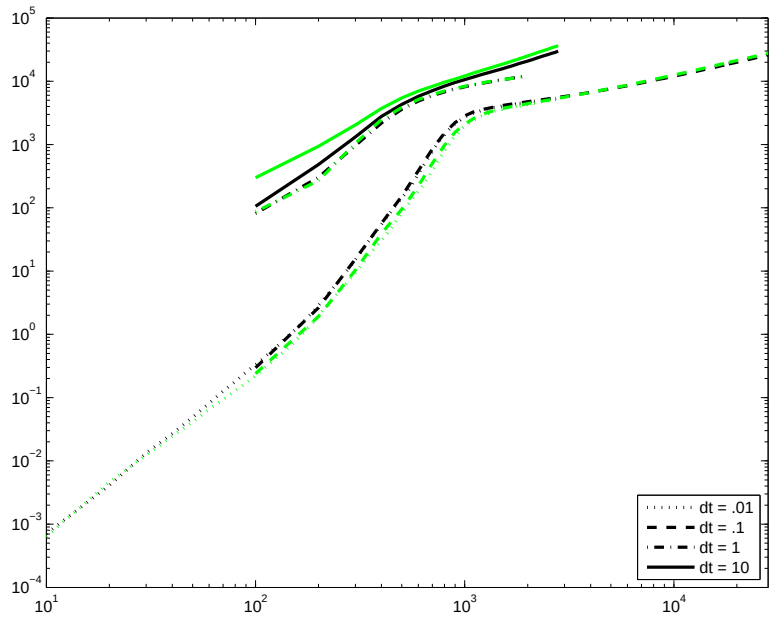

Figure 10. Simulations on a structured (black) and unstructured (green) mesh both with $h \leq 4\lambda$ are compared. Ion temperature is shown using time step sizes $\{\frac{1}{100}, \frac{1}{10}, 1, 10\}$.

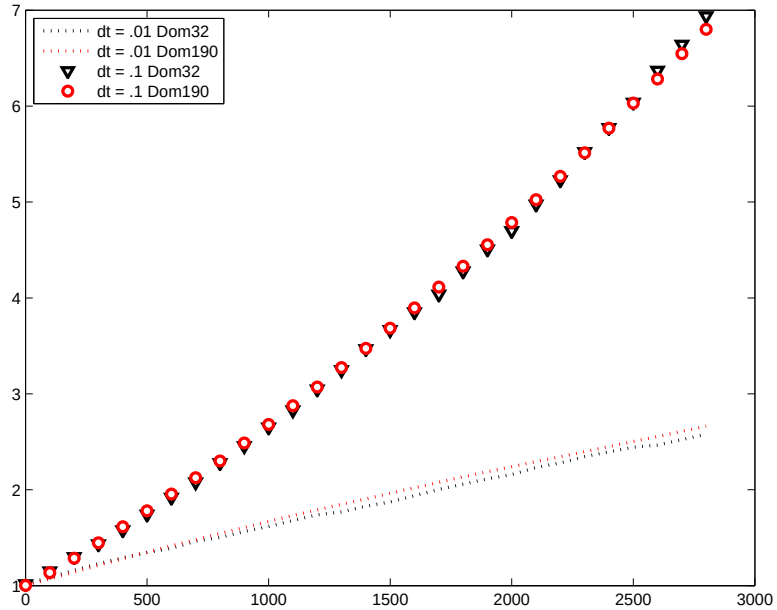

Figure 11. The spatially averaged *electron* temperature is shown for both an unstructured and a structured mesh both with $h \leq \lambda$. Temperature and time are both shown on *linear* scales. Time integration using time step size $\frac{1}{100}$ is shown in dotted (structured) and dashed (unstructured). Time integration using time step size $\frac{1}{10}$ is shown with circles (structured) and diamonds (unstructured). Simulations with time step size $\frac{1}{100}$ used 280000 time steps. Simulations with time step size $\frac{1}{10}$ used 28000 time steps. Both simulations heat up severely.

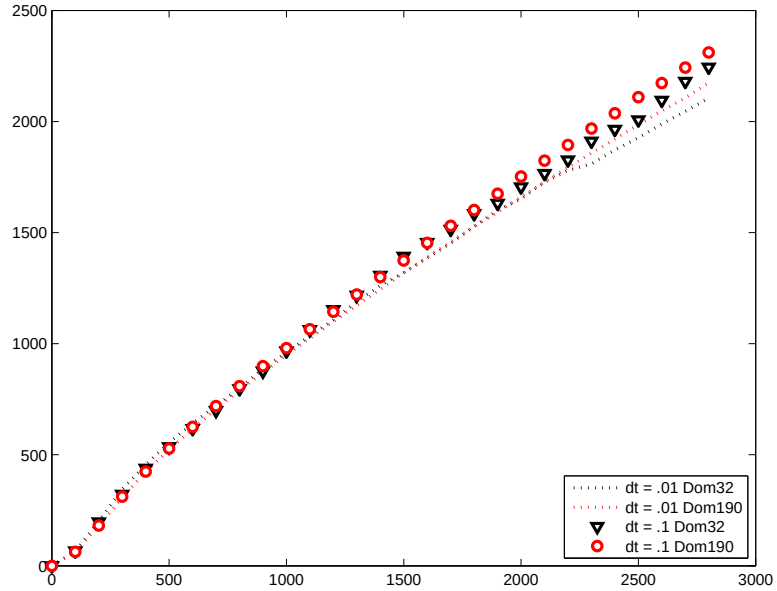

Figure 12. The spatially averaged *ion* temperature is compared on a structured (dots,solid) mesh to an unstructured(dash, dash-dot) mesh, both satisfying the Debye stability criterion $h \leq \lambda$. Time integration with time step size $\frac{1}{100}$ (dots structured, dashes unstructured) is compared to time integration with time step size $\frac{1}{10}$ (solid structured, dash-dot unstructured).

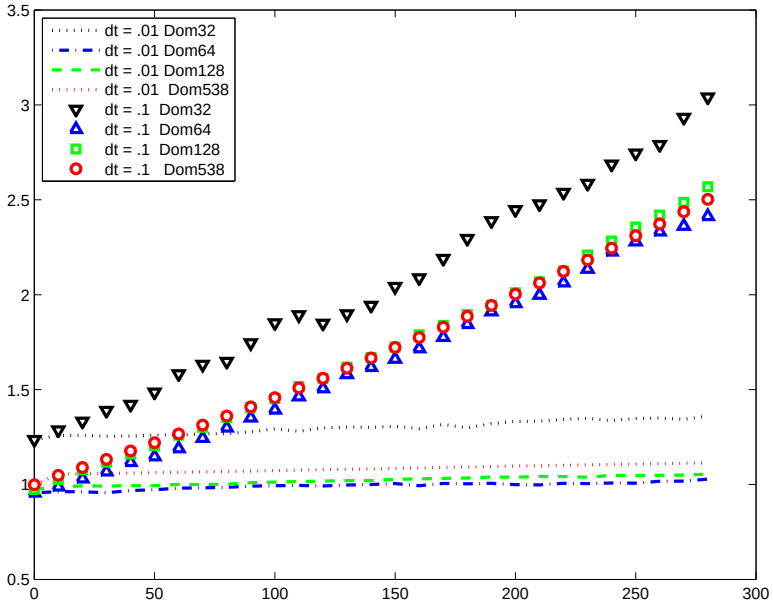

Figure 13. The spatially averaged *electron* temperature is shown for a structured mesh both with $h \leq \lambda/4$. Temperature and time are both shown on *linear* scales. Four n by n meshes are used with $n = 32, 64, 128, 538$. The spatially averaged electron temperature with step size $\frac{1}{10}$ and 2800 time steps is shown with down triangles, up triangles, squares and circles for the small,medium, large and extra large domains respectively. Time integration using time step size $\frac{1}{100}$ and 28000 time steps is shown in dotted, dash-dot, dashed and dotted lines for the small,medium, large and extra large domains respectively. With step size $\frac{1}{100}$, the simulation appears stable. On the small mesh, the initial temperature is higher than requested, due to the small total number of particles.

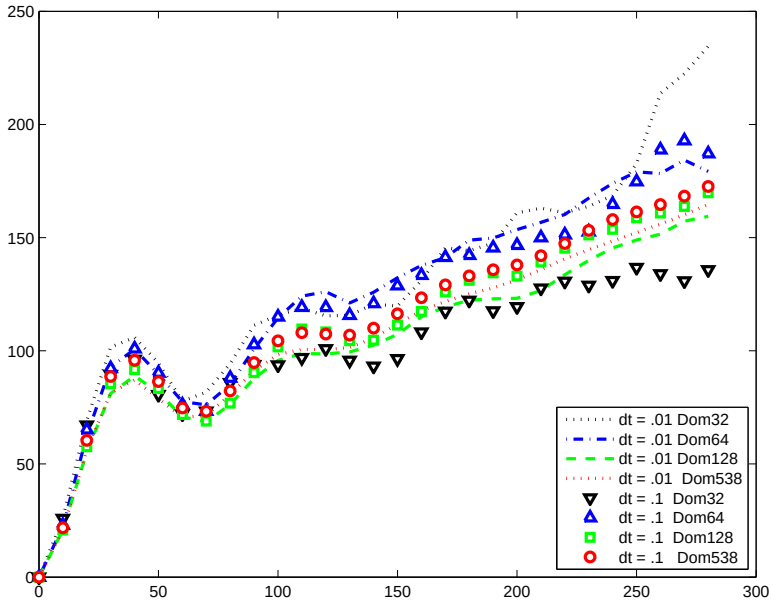

Figure 14. The spatially averaged *ion* temperature is compared on four Debye stable structured meshes, $h \leq \lambda/4$. The meshes are n by n for $n = 32, 64, 128, 538$. Time integration with time step sizes $\frac{1}{10}$ produces similar spatially averaged ion temperatures to time integration with time step size $\frac{1}{100}$. .

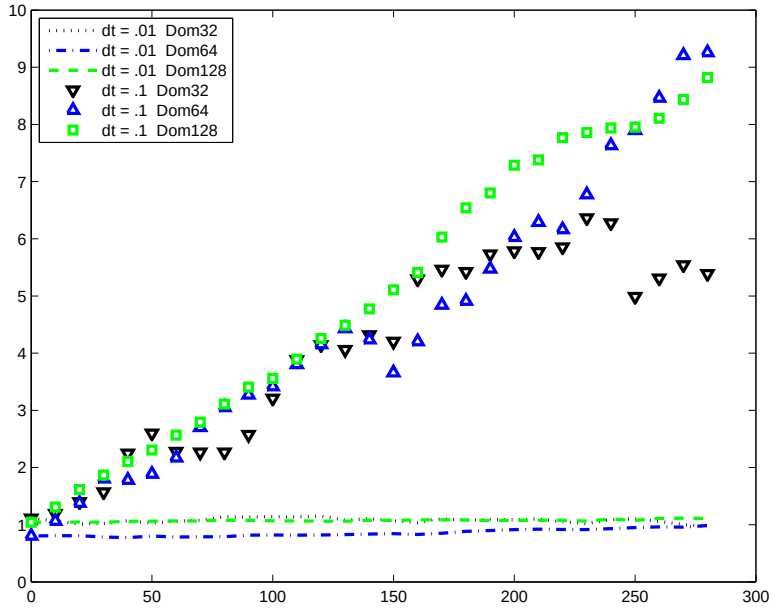

Figure 15. The spatially averaged *electron* temperature is shown for structured meshes each with $h \leq \lambda/16$. Three n by n structured meshes are comppared with $n = 32, 64, 128$. Temperature and time are both shown on *linear* scales. The dotted lines show Time integration using time step size $\frac{1}{100}$ and 28000 time steps. Time integration using time step size $\frac{1}{10}$ and 2800 time steps is shown with down triangles ($n = 32$), up triangles ($n = 64$), and squares ($n = 128$). Time integration using time step size $\frac{1}{100}$ is shown with dotted lines ($n = 32$), dash dot ($n = 64$), and dashed lines ($n = 128$). The $n = 32$ mesh is much too small. The time step $\frac{1}{10}$ is too large, violating a CFL condition. The simulation on the $n = 128$ mesh with the small time step is stable.

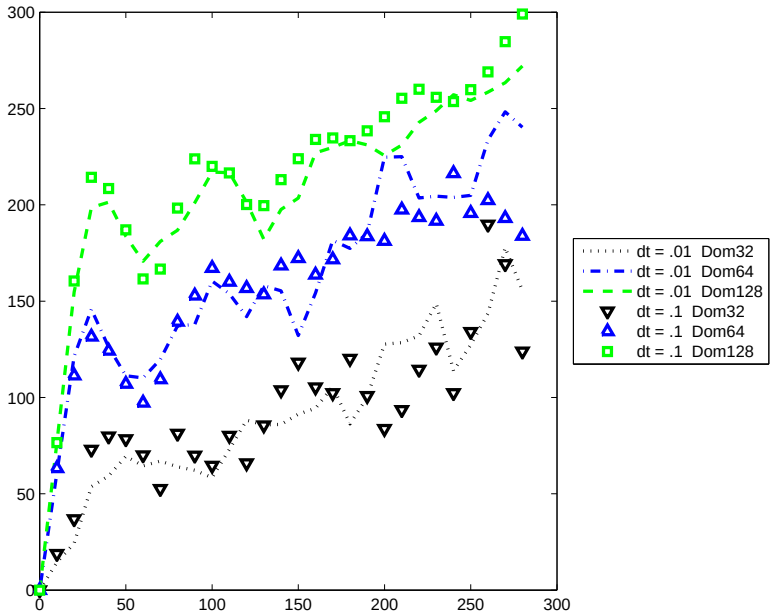

Figure 16. The spatially averaged *ion* temperature is compared on n by n structured meshes with $h \leq \lambda/16$ for $n = 32, 64, 128$. Temperature and time are both shown on *linear* scales. The time step is small enough to resolve the ions. However, the number of particles is too low to produce mesh independent trajectories over long times.

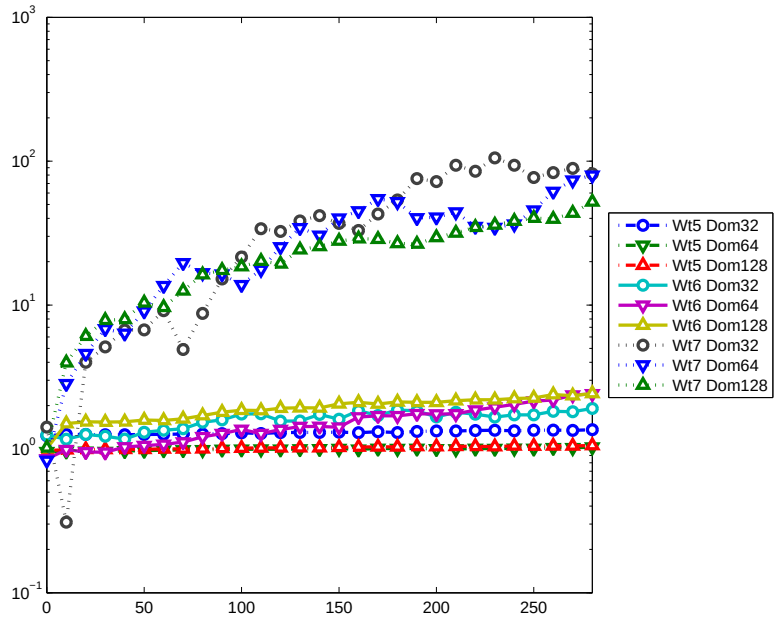

Figure 17. The spatially averaged electron temperature is shown for simulations using meshes are Debye stable, $h \leq \lambda/4$. There are 9 simulations, with three different numerical particle weights $10^{5:7}$ and three different domain sizes $h2^{5:7}$. The simulations with weight 10^5 on the 64×64 and 128×128 domains do not heat up.

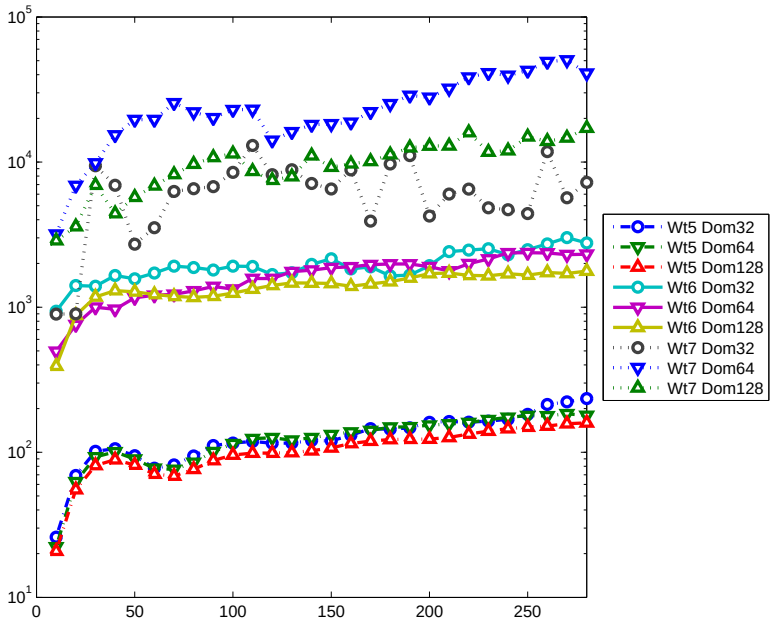

Figure 18. The spatially averaged *ion* temperature is shown on Debye stable meshes, $h \leq \lambda/4$, for different numerical particle weights, $10^{5:7}$, and different domain sizes $h2^{5:7}$.

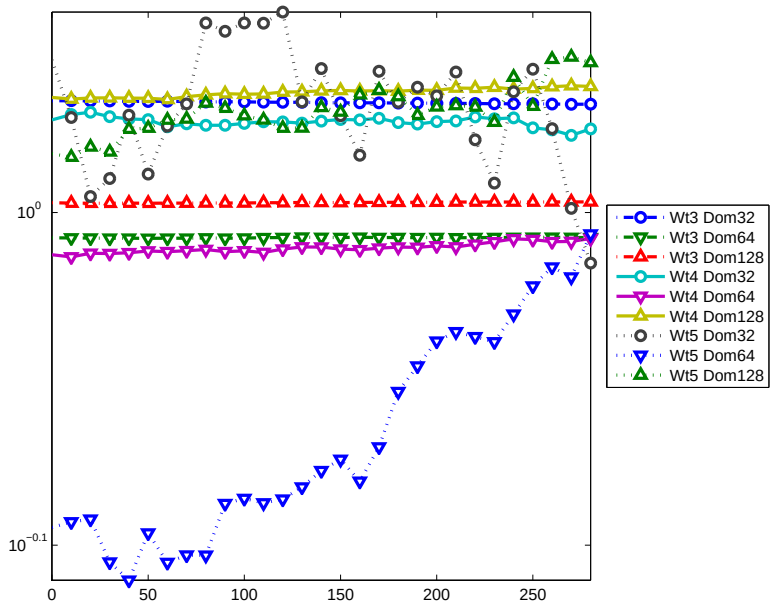

Figure 19. The spatially averaged electron temperature is shown for meshes that are Debye stable, $h \leq \lambda/16$, but for numerical particle weights $10^{3:5}$ and for domain sizes $h2^{5:7}$.. The simulations with weight 10^3 on the 64×64 and 128×128 domains do not heat up.

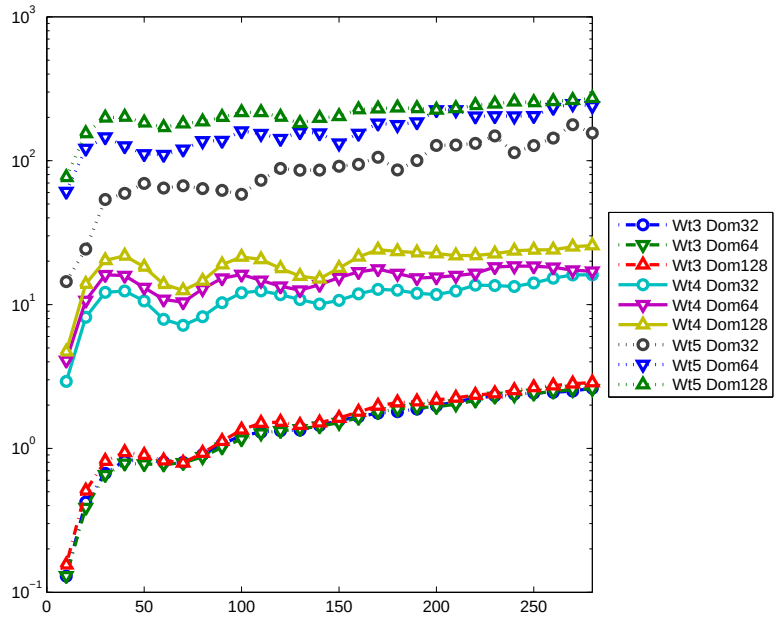

Figure 20. The spatially averaged ion temperature is shown for meshes that are Debye stable, $h \leq \lambda/16$, but for numerical particle weights $10^{3:5}$ and domain sizes $h2^{5:7}$. For the apparently stable simulations, weight 10^3 and either 64×64 or 128×128 domains, the ion temperature is lower here than with $h = \lambda/4$.

Heating Rates

> It is well known that particle models of a plasma grossly exaggerate the collision rate of the system they simulate, due to the small number of particles used in the model compared with the real system. Such models also invariably heat up due to the presence of stochastic error fields arising from the finite size of the space and time steps [6].

In this section a heating rate for the spatially averaged temperature $T(t)$ is determined. If $T(t) = T(0)e^{at}$, then the heating rate is a. After preliminary comments, Figures 21, 22, and 23 are presented showing the results of numerical experiments in which the initial temperature is varied. These figures show that the heating rate decreases as the initial temperature increases. Figures 24, 25 and 26 compare the heating rate to the mesh size, CFL number and an estimate of the overall computational cost. The section concludes with Table 4 documenting the mesh resolution, CFL condition, numerical particle density and the heating rates the rates the 20 most stable simulations of the test problem, all starting from the same initial λ, ω_p, temperature and electron density.

For (structured) PIC, the heating rate depends on mesh size, Debye length, and time step size. For sufficiently small time step size, the growth rate varies with dx/λ. Due to $\lambda = C\sqrt{T}$, non-trivial models of $T(t)$ are anticipated. For many simulations $T(t)$ is nearly a curve of the form $T(0)e^{at-bt^2}$ with positive a and b over specific interval $[0,t_f]$. This narrows the uncertainty in the growth rate, a, by more precisely fitting $T(t)$ over meaningful time spans, $t_f \leq \frac{a}{2b}$.

The interpolation scheme used for all simulations is $T(t) \approx T_{\min}e^{at-bt^2}$. For reasons not known to the author, when $dx \ll \lambda$, $T(t)$ sometimes decreases for the first few time steps. The interpolation scheme uses the minimum observed value $T_{\min}$ instead of T_0.

Table 4 the growth rates for all the heating curves (for structured meshes) the were approximately exponential. The CFL condition is $dtv_{Th} \leq Cdx$ for C to be determined. While small values of dx/λ and dtv_{Th}/dx are necessary for a small growth rate, these values do not alone predict the growth rate. Recall that $dtv_{Th} \leq Cdx = dt\frac{\omega_p}{2}\frac{\lambda}{dx}$. No simple universal model predicting the growth rate is apparent.

The runs with shown in Figure 11 with $dx = \lambda$, numerical particle weight 10^5, on the 32 by 32 domain with time step size .1 and .01 are also included. These simulations have 1.38 particles per element. Heating curves for the simulations with larger particle weights (fewer numerical particles) were not of the form e^{at-bt^2}, and no rate could be determined.

Runs with $dx = \lambda/4$ with high numerical particle weights were also too poorly resolved to determine a heating rate. The two that are the best resolved are included. These have weight 10^5 and the mesh is 128 by 128. The time step sizes are .1 and .01 respectively. Additional simulations with numerical particle weights weights $10^{3:4}$ were included for completeness.

Runs with $dx = \lambda/16$ and weights $10^{3:5}$ are included. The 32 by 32 domain is adequate for the growth rates above .2 at time step size .1, even though the domain diameter is only 2λ. The 128

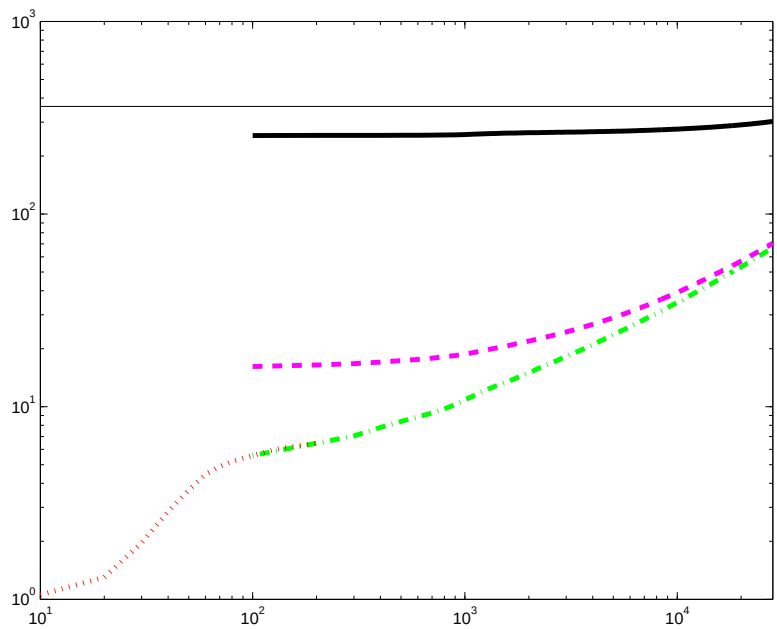

Figure 21. Spatially averaged electron temperature is studied for simulations with initial temperatures $T(0)$ of $\{1, 16, 256\}$ electron volts. Each simulation uses the same structured mesh, time step size $1/10$ (in units of $2/\omega$) and numerical particle weight 10^5. For the simulation with $T(0) = 1$ eV, the mesh is Debye unstable, $h = 16\lambda$. Three simulations take 282000 time steps, with output stride 1000. A fourth simulation with $T(0) = 1$ eV with output stride 100 and 2000 time steps is also shown.

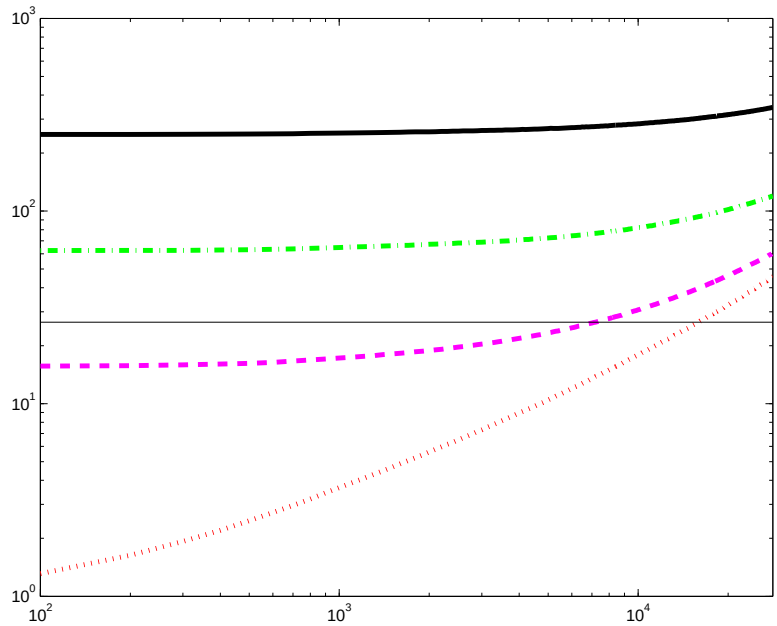

Figure 22. Spatially averaged electron temperature is studies for simulations with initial temperatures $T(0)$ of $\{1, 16, 64, 256\}$ electron volts. Each simulation uses the same structured mesh, time step size $1/10$ (in units of $2/\omega$) and numerical particle weight 10^5. For the simulation with $T(0) = 1$ eV, the mesh is Debye unstable, $h = 16\lambda$. The simulations take 282000 time steps, with output stride 1000.

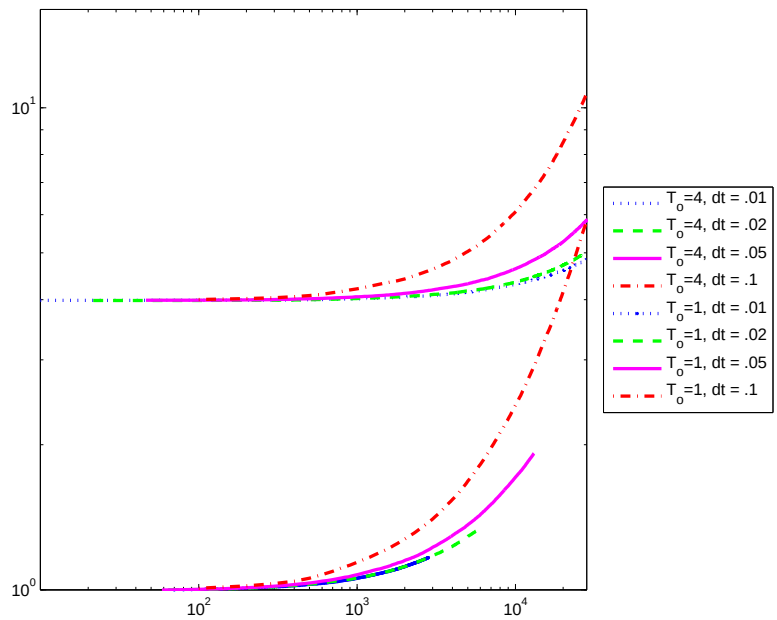

Figure 23. The spatially averaged electron temperature is shown for simulations using meshes with $h \leq \lambda$. There are 8 simulations. Each simulation has numerical particle weights 10^4 and domain size $32h$. The time step sizes used are $\{.01, .02, .05, .1\}$ in units of $2/\omega$. The initial temperature is either one or four electron volts. The simulations starting from $1eV$ use 282000 time steps. The simulations starting from $4eV$ are run out to 10 uSec.

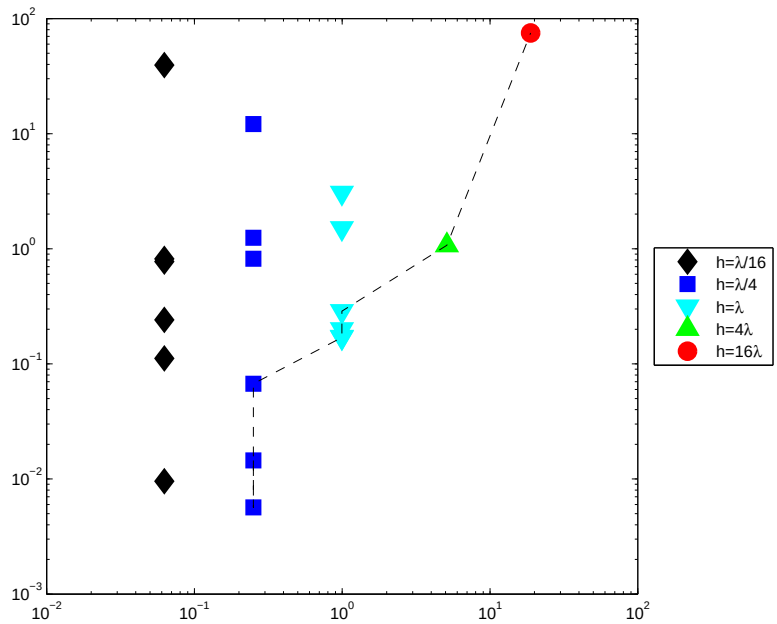

Figure 24. The heating rate per microsecond is shown (y-axis) versus dx/λ for 20 simulations with the smallest heating rates. The circle, up triangle, down triangle, square and diamond denote simulations with dx/λ of 16, 4, 1, $1/4$ and $1/16$ respectively. The dotted line shows the minimum heating observed as a function of dx/λ. The parameters λ, ω_p, temperature and electron density are the same initially. The corresponding parameter and heating values are documented in Table 4.

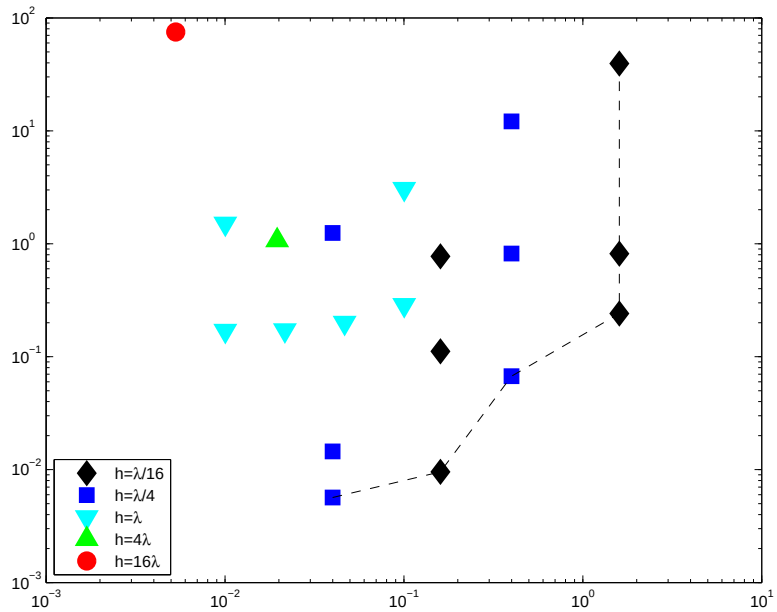

Figure 25. The heating rate per microsecond (y-axis) is compared to the Courant number $v_{Th}dt/(dx\lambda)$ for 20 simulations with the smallest heating rates. The circle, up triangle, down triangle, square and diamond denote simulations with dx/λ of 16, 4, 1, $1/4$ and $1/16$ respectively. Each simulation uses the same λ, ω_p, temperature and electron density (see Nomenclature). The corresponding parameter and heating values are documented in Table 4. The dotted line shows the minimum heating observed as a function of the Courant number. The maximum Courant number among the 20 best simulations is 1.6. On the other hand, further reductions in Courant number do not appear to significantly reduce heating.

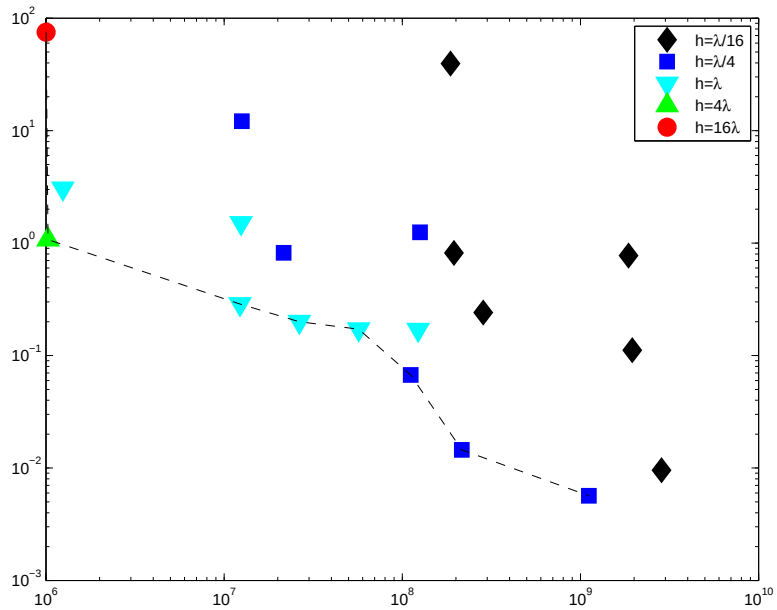

Figure 26. The heating rate per microsecond is shown (y-axis) is compared to the "cost", given by $(K+P)/dt$ where K is the number of elements, P is the total number of numerical particles, and dt is the time step size measured in units of $\omega_p/2$. The values of K and P correspond to the $1cm^2$ domain; these values are estimated for the scaled down domains. The circle, up triangle, down triangle, square and diamond denote simulations with $dx/lambda$ of $16, 4, 1, 1/4$ and $1/16$ respectively. Each simulation uses the same values of λ, ω_p, temperature and electron density. The dotted line interpolates the most effective parameters as a function of "cost".

Table 4. The growth rates for the spatially averaged temperature for structured meshes are tabulated for the 20 simulations shown in Figures 24-26. The parameters λ, ω_p, temperature and electron density are the same initially. The third column is the number of numerical particles per element. Increasing the number of numerical particles per cell consistently reduces the heating rate.

dx/λ	$dt\ v_{Th}/dx$	NumPartPerCell	Growth Rate
$1.88e+01$	$5.31e-03$	$5.00e+02$	$7.50e+01$
$5.11e+00$	$1.96e-02$	$3.65e+01$	$1.07e+00$
$9.95e-01$	$1.00e-02$	$1.60e+01$	$1.70e-01$
$9.95e-01$	$2.16e-02$	$1.60e+01$	$1.71e-01$
$9.95e-01$	$4.66e-02$	$1.60e+01$	$2.00e-01$
$9.95e-01$	$1.00e-01$	$1.60e+01$	$2.87e-01$
$9.95e-01$	$1.00e-02$	$7.24e-01$	$1.51e+00$
$9.95e-01$	$1.00e-01$	$7.24e-01$	$3.06e+00$
$2.50e-01$	$4.00e-02$	$8.64e-02$	$1.25e+00$
$2.50e-01$	$4.00e-01$	$8.64e-02$	$1.21e+01$
$2.50e-01$	$4.00e-02$	$8.64e-01$	$1.45e-02$
$2.50e-01$	$4.00e-01$	$8.64e-01$	$8.19e-01$
$2.50e-01$	$4.00e-02$	$8.63e+00$	$5.66e-03$
$2.50e-01$	$4.00e-01$	$8.63e+00$	$6.72e-02$
$6.25e-02$	$1.60e+00$	$5.37e-01$	$2.41e-01$
$6.25e-02$	$1.60e+00$	$5.27e-02$	$8.18e-01$
$6.25e-02$	$1.60e+00$	$5.86e-03$	$3.94e+01$
$6.25e-02$	$1.60e-01$	$5.40e-01$	$9.54e-03$
$6.25e-02$	$1.60e-01$	$5.40e-02$	$1.12e-01$
$6.25e-02$	$1.60e-01$	$5.40e-03$	$7.73e-01$

by 128 domain is needed for runs with time step size .01.

Conclusions

Our primary conclusion concerns the Debye mesh constraint. The plasma Debye length imposes a constraint on the maximum element diameter, h. All simulations with $h \geq \lambda$ heated up. No simulation with $h \leq \lambda/4$ and time step size that satisfies the CFL condition, equation (5), heated up. For $h \approx 16\lambda$, it was unclear whether the heating would level off once the spatially averaged temperature had increased the effective Debye length to h. For $h \approx 4\lambda$ and $h \approx \lambda$, the heating continued even after the effective Debye length exceeded h.

A plasma in a postage stamp problem, 1cm by 1cm, with specular boundary conditions is considered. The numerical particle weight are always $\geq 10^3$. The meshes are all quasi-uniform. Heating rates for the spatially averaged electron temperature, $T(time) \approx T(0)exp(rate * time)$ are determined. The precise relationship between the heating rate and the input parameters is complex. If $dx < \lambda$, $dtv_{Th} < .2dx$, and the number of particles is adequate, heating rates near $1/10$ are observed. If dx is nearly λ, then a small time step and a small weight may be needed to achieve the desired heating rate. Meshes with $dx \leq \lambda/4$ appear to be the most economical overall.

Similar results for the PIC method are reported in [6]. A natural follow on problem is to make more detailed comparisons between PIC and unstructured PIC.

Other conclusions and observations are ordered chronologically.

Simulation Time. The time step size influences the total simulation time in many ways. Simulations with fixed number of time steps and the largest proposed time step sizes are computationally prohibitive.

If a simulation is unstable, and heats up disastrously, the computation time also increases. For example Table 5 shows the total run times for eight simulations, all with 2000 time steps. The domain is $[0,1]^2$ in all cases. The element size and the time step size vary. The explosion of computation time is more severe here than it would be for a structured PIC code. The corresponding times for the linear systems are given in Table 6 A structured PIC code would, at the end of a time step, determine the new particle position, and then look up the new element that contains the particle. An unstructured PIC code moves particles only using element adjacency. An option if simulation time has ballooned is to visualize the simulation so far, and manually terminate the simulation if it has become unstable.

In initial set of experiments included unstable simulations that were not practical. Recall that there are 86400 seconds in a day. Ignoring the time spent pushing particles changes the bottle necks.

In Table 6 the solve time per element actually decreases as number of elements increases. The linear solver used in all of these experiments (for consistency) is efficient only for sufficiently large linear systems.

Physical Time. For the simulations, the time interval of interest turned out to be roughly one μ second. This is 28208 time steps at $dt = .2/\omega$, that is $dt = 1/10$ in units of $2/\omega$. The plasma

Table 5. Particle calculation time (seconds) for simulations with 2000 time steps

Mesh	Structured?	num core	$dt = .01$	$dt = .1$	$dt = 1$	$dt = 10$
squareL16	Yes	2	317	314	724	36414
squareL16U	No	2	303	316	534	44750
squareL4	Yes	4	182	184	944	94624
squareL4U	No	4	165	181	919	85146
squareL	Yes	8	164	190	6606	620173
squareLU	No	8	180	200	6828	na
squareLm4	Yes	8	403	505	na	na
squareLm4U	No	8	180	200	na	na

Table 6. Solve time (seconds) for simulations with 2000 time steps

Mesh	Structured?	num core	$dt = .01$	$dt = .1$	$dt = 1$	$dt = 10$
squareL16	Yes	2	70	68	62	63
squareL16U	No	2	65	65	64	63
squareL4	Yes	4	97	93	83	310
squareL4U	No	4	83	88	78	183
squareL	Yes	8	342	358	388	436
squareLU	No	8	370	383	404	na
squareLm4	Yes	8	10224	7526	na	na
squareLm4U	No	8	9281	9419	na	na

frequency is derived under the assumption that the plasma state is a small sinusoidal oscillation about a uniform plasma. For the simulations that were stable (did not heat up), the oscillations in the spatially averaged ion temperature were within an order of magnitude of the ion plasma frequency.

Time step stability. The time step size for Stormler - Verlet is bounded in terms of a different mesh dependent frequency, $\omega_{\max}$, the maximum normal mode frequency of the linearized system of equations [4] If $dt > 2/\omega$, the time integration is unstable. For $dt \gg 2/\omega$, the growth rate per iteration is large and the particle positions are mesh dependent after only a couple iterations. We have not yet calculated ω_max.

Numerical particle per cell. Increasing the average number of numerical particles per cell consistently reduces the heating rate. This is in contrast to decreasing the time step size and the mesh size, which are only helpful near to the critical thresholds.

Another issue that is sensitive to the number of particles per cell is the information determined from a simulation. Element based quantities are accurate if the average number of numerical particles per element is high. Otherwise the initial spatially averaged temperature is typically not the requested temperature. We observed accurate temperatures for at least 33 particles per element, and inaccurate temperatures for less than 1.4 particles per element.

Debye mesh constraint. The plasma Debye length imposes a constraint on the maximum element diameter, dx. In Figure 24 the most efficient simulations have dx/λ of either 1 or $1/4$, depending on precisely how much heating is acceptable.

In Table 1 as the number of elements increases, the unstructured meshes use more elements to achieve a given maximum element diameter than the structured meshes. On the other hand, for example in Figure 11, the maximum element diameter appears to be the best predictor of heating. This issue needs to be studied more thoroughly.

Numerical particle weights. The plasma parameter, $N_D = (4/3)\ \pi\lambda^3 n = 1.4e+4$ is the number of physical particles in a sphere of radius λ_D. Weights below N_D are needed in many simulations.

In these experiments numerical heating is defined subjectively. Figure 15 shows no heating with numerical particle weight 10^5 and time step size $1/100$. On the other hand Figure 19 shows heating with numerical particle weight 10^5 and time step size $1/100$. In Figure 15 the simulation with time step size $1/10$ heats severely, and by comparison there is no heating at the smaller time step. And decreasing the particle weight does reduce the heating, as in Figure 19. The heating in Figure 19 appears to reflect the resolution of the particle distribution. Stable simulations are insensitive to the numerical particle weight as long as there is an adequate number of numerical particles. For the simple problem considered here, around 100 numerical particles suffice.

For the variable domain sizes with particle number density and constant particle weight, the domain size is proportional to the total number of particles. Table 7 gives the number of particles per species for the meshes with either $h = \lambda/16$ or $h = \lambda/4$. For $h = \lambda/4$, 178 particles per species are enough for the electron temperature, to be constant, but at least 709 particles per species are needed to resolve the ion heating.

Table 7. Reduced or scaled down domains were used in simulations with mesh size $dx \ll \lambda$. For fixed numerical particle weight, the reduction in the total number of particles becomes significant.

dx	*Weight*	*Domain32*	*Domain64*	*Domain128*
$\lambda/16$	10^3	1094	4422	17684
$\lambda/16$	10^4	107	428	1716
$\lambda/16$	10^5	12	46	177
$\lambda/4$	10^5	178	709	2830
$\lambda/4$	10^6	18	71	283
$\lambda/4$	10^7	2	7	28

Benign warning During the weight 10^6 and 10^7 runs with < 28 numerical particles, the warning message "Aztec status AZ_loss: loss of precision" is printed to the screen at certain time steps. It is surprising that in this case these warning messages are benign. The warning messages are due to the simple convergence criterion of the iterative linear solver for the potential equation. At each step, the initial guess is the solution from the previous iteration. In simulations with so few numerical particles on some time steps no particles leave their elements. On such time steps, the right-hand side is the same as at the previous iteration. The initial guess is an approximate solution. However, the iterative method is asked to decrease the relative residual norm by a second factor of 10^{-9}, for a total of 10^{-18}, which is not in general possible in double precision (typically machine precision is 2^{-54}).

References

[1] C. K. Birdsall and A. B. Langdon. *Plamsa Physics Via Computer Simulation*. McGraw-Hill, 1985.

[2] A. Draganescu, T. F. Dupont, and L. R. Scott. Failure of the discrete maximum principle for an elliptic finite element problem. *Math. Comp.*, 74:1–23, 2005.

[3] N.A. Gatsonis and A. Spirkin. A three-dimensional electrostatic Particle-In-Cell methodology on unstructured Delaunay-Voronoi grids. *J. Comp. Phys.*, 228(10):3742–3761, 2009.

[4] B.B. Godfrey. Canonical momenta and numerical instabilities in particles codes. *J. Comp. Phys.*, 19:58–76, 1975.

[5] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Taylor & Francis Group, 1988.

[6] R.W. Hockney. Measurements of collision and heating times in a two-dimensional thermal computer plasma. *J. Comp. Phys.*, 8(1):19–44, 1971.

[7] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, Inc., 1963.

[8] H. R. Lewis. Energy-conserving numerical approximations for Vlasov plasmas. *J. Comp. Phys.*, 6:136–141, 1975.

[9] H. Okuda. Nonphysical instabilities in plasma simulation due to small $\lambda_d/\delta x$. *Fourth Conference on the Numerical Simulation of Plasma*, pages 511 – 525, 1970.

[10] A. Spirkin and N.A. Gatsonis. Unstructured 3D PIC simulations of the flow in a retarding potential analyzer. *Comput. Phys. Comm.*, 164:383–389, 2004.

[11] J.-S. Wu, K.-H. Hsu, F.-L. Li, C.-T. Hung, and S.-Y. Jou. Development of a parallelized 3D electrostatic PIC-FEM code and its applications. *Comput. Phys. Comm.*, 177:98–101, 2007.

A Physical Constants

In this section, the conversion of temperature to Debye length, and electron density to plasma frequency are worked out in fine detail.

$$\text{debyeLength}^2 = \lambda^2 = \frac{\text{permittivity boltzmann temperature}}{\text{numberDensity elementaryCharge}^2}$$

where

$$\text{permittivity} = \varepsilon_0 = \text{const ampere second}/(\text{volt meter}),$$

for

$$\text{const} = 8.854187817 \times 10^{-12}.$$

The plasmas simulated here are at

$$\text{temperature} = T = 11605 \text{ Kelvin}$$

$$\text{boltzmann temperature} = \text{electronVolt} = eV = \text{elemChargeC joule},$$

where elemChargeC $= 1.60217646\ 10^{-19}$ is the elementary charge in Coulombs.

$$\text{elementaryCharge} = e = \text{elemChargeC ampere second}, \quad \text{volt} = \text{joule}/(\text{ampere second})$$

If numberDensity, n, is measured in cm^3, then $\lambda = \sqrt{\text{const}/(100n \text{ elemChargeC})}$. For $n = 10^{10}\text{cm}^{-3}$, there holds $\lambda = .74339424 \times 10^{-2}$ cm.

$$\text{plasmaFrequency}^2 = \frac{\text{electronDensity}(\text{elementaryCharge}^2)}{\varepsilon_0 \quad \text{electronMass}}$$

Expanding volts and elementary charge as above,

$$\varepsilon_0 = \text{const}\frac{\text{amp sec}}{\text{volt meter}} = \text{const}\frac{\text{amp}^2\text{sec}^2}{\text{joule meter}}$$

$$\text{elementaryCharge}^2/\varepsilon_0 = \frac{\text{elemChargeC}^2}{\text{const}}\text{joule meter}$$

$$\text{plasmaFrequency}^2 = \frac{\text{electronDensity}}{\text{electronMass}}\frac{\text{elemChargeC}^2}{\text{const}}\text{joule meter}$$

Using joule $=$ kg $\frac{\text{meter}^2}{\text{sec}^2}$, for electronMass in kg and electronDensity in meters^3, plasmaFrequency has units $1/sec$. In cgs units, plasmaFrequency is measured in micro seconds. electronDensity = $10^{10}\text{cm}^{-3} = 10^{16}\text{meter}^{-3}$. Substituting electronMassKg $= 9.10938188\ 10^{-31}$,

$$\text{plasmaFrequency} = \left((\text{electronDensity}/\text{electronMassKg})(\text{elemChargeC}^2)/\text{const})10^{-6}\right)^{1/2}$$

$$= 5.6415\ 10^3/\mu\text{sec}$$

B Conservation Laws

The equations modeling an electrostatic plasma presented in section are presented in a more general framework. Issues related to the evaluation of discretization methods, such as conservation laws and verification, are emphasized.

The simulation methods discussed here involve a finite number of numerical particles at locations $\{x^i\}$ with velocities $\{v^i\}$. The plasmas considered here have ions (protons) of charge e and electrons of charge $-e$. The particle weight is the number of particles per numerical particle.

The traditional (§5.5 in [5] and §10.5 of [1]) starting point of the analysis is the Action integral for Vlasov's equation with potential Φ, and Lagrangian $\mathscr{L}$,

$$I[\Phi, \mathbf{x}] = \int_{t_0}^{t_1} \mathscr{L} dt \tag{6}$$

Homogeneous Dirichlet boundary conditions apply on the volume element in phase space. The particle Lagrangian is

$$\mathscr{L} = \frac{\varepsilon}{2}\langle E, E\rangle + \frac{1}{2}\sum_{i=1}^{n_p} m^i |v^i|^2 - q^i \Phi(x^i) \tag{7}$$

The charge q^i of the ith numerical particle is product of the particle weight and $\pm e$. The charge density field ρ is ultimately a sum of delta functions at the particle positions each weighted by their charge. Gauss' Law, relating the electrostatic potential ϕ and the charge density ρ, equation (2), is the Lagrange equation for variations of (7) with respect to Φ. Varying (7) with respect to position produces Coulomb's Law, equation (3). A particle evolving according to Coulomb's Law conserves its energy,

$$\frac{d}{dt}\left(\frac{1}{2}m^i|v^i|^2 + q^i\Phi(x^i(t))\right) = 0.$$

The kinetic energy of the particles is essentially electron temperature,

$$\frac{m_e}{2k_B}\sum_{anions} |v^i|^2$$

Constants accounting for units and particle weight have been omitted for clarity. If the potential energy is less than the maximum kinetic energy, then the temperature plots are essentially the energy.

Other equations for particles, such as the potential energy of particles in free space

$$U_{\text{particle}} = \frac{1}{4\pi\varepsilon}\sum_{i<j}\frac{q_i q_j}{r_{ij}},$$

motivate field equations.

The classical argument (§6.9 of [7]) for momentum conservation applies here using Coulomb's Law in the form $dp^{(i)}/dt = q^{(i)}E$, and only that the charge density field ρ averaged over each finite

element K must satisfy equation (1). The sum of the momenta, p_V, of all the particles in a given volume V evolves according to

$$\frac{dp_V}{dt} = \int_V \rho(x)E(x)d^3x. \tag{8}$$

In the discrete case, V is a union of finite elements. Within each element E is constant, and momentum is conserved as long as no particle trajectories intersect ∂V. It is traditional to express the discrete equations in terms of the shape functions and particularly the discrete Green's functions [2]. The continuum case can be expressed in conservation form using the Maxwell stress tensor, $\mathbf{T}$. Thinking of E as a column vector and of divergence as transforming matrices to row vectors, in the electrostatic ($B = 0$) case $\mathbf{T} = EE^T - \mathbf{I}\frac{1}{2}|E|^2$, where $\mathbf{I}$ denotes a 3 by 3 identity matrix, and

$$\int_V \rho(x)E(x)d^3x = \int_V \mathrm{div}\varepsilon\mathbf{T}d^3x = \int_{\partial V} \varepsilon\mathbf{T}\cdot dS$$

This suggests many different ways to check conservation laws. Few of these have been pursued. One that stands out is to account for particles crossing element boundaries. A framework that can keep track of crossings was worked out in the heyday of PIC methods [8].

Based on U_{particle}, in general the potential energy within a volume V

$$U = 1/2\int_V \rho\phi = \frac{\varepsilon}{2}\int_V \nabla\cdot(E\phi) - E\nabla\phi = \frac{\varepsilon}{2}\int_{\partial V} \phi E\cdot dS + \int_V \frac{\varepsilon}{2}|E|^2.$$

which depends on the electrostatic energy density $u = \frac{\varepsilon}{2}|E|^2$ and a boundary term. The current density (which here is displacement current) $J = -\partial_t \varepsilon E$ satisfies the continuity equation $\nabla\cdot J = -\partial_t\rho$. And due to $B = 0$ Poynting's Theorem states that $\partial_t u = -\langle E, J\rangle$.

DISTRIBUTION:

1 MS 0899 Technical Library, 9536 (electronic)

Sandia National Laboratories