# A Particle in Cell Corrected Approach to Direct Sum

Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, and Madie Wilkin
Adviser: Dr. Andrew Christlieb
Co-advisers: Eric Wolf and Justin Droba

Michigan St. University

July 23, 2014

# The N-Body Problem



- Solar systems

# The N-Body Problem



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- Solar systems
- Interacting charges, gases and plasma

# The N-Body Problem



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- Solar systems
- Interacting charges, gases and plasma
- We want to solve the system of ODE's given by:

$$\frac{dx_i}{dt} = v_i \qquad \frac{dv_i}{dt} = a_i = \frac{F_i(x_1, \ldots, x_n)}{m_i}$$

for $i = 1, \ldots, n$.

# Traditional Solutions



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- Direct Sum
- Particle Mesh
- Particle-Particle Particle-Mesh

- The net force on particle $i$ is:

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} q_i E_{j,i}$$

- The net force on particle $i$ is:

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} q_i E_{j,i}$$

- $E_{j,i}$ is the electric field from particle $j$ at the location of particle $i$ and $q_i$ is the charge

- The net force on particle $i$ is:

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} q_i E_{j,i}$$

- $E_{j,i}$ is the electric field from particle $j$ at the location of particle $i$ and $q_i$ is the charge
- To calculate the force on all of the particles, we must repeat this summation for each of the $n$ particles

- The net force on particle $i$ is:

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} q_i E_{j,i}$$

- $E_{j,i}$ is the electric field from particle $j$ at the location of particle $i$ and $q_i$ is the charge
- To calculate the force on all of the particles, we must repeat this summation for each of the $n$ particles
- The problem with direct sum is that it is $\mathcal{O}(N^2)$

# Goal

- Find efficient ways to solve the n-body problem.

# Goal



- Find efficient ways to solve the n-body problem.
- Improve efficiency from $\mathcal{O}(n^2)$ to $\mathcal{O}(n) + \mathcal{O}(k^2)$ where $k << n$ and $k$ is the number of particles per cell.

$$E_{j,i} = \frac{q_j}{\epsilon_0}\mathcal{G}(x_j|x_i)$$

$$F_{j,i} = q_i * E_{j,i}$$

Table: Theoretical Time Comparison (3600 MHz computer)

| $N$ | $\mathcal{O}(N^2)$ | $\mathcal{O}(N)$ |
|---|---|---|
| 10 | $10^{-8}$ sec | $10^{-9}$ sec |
| $10^3$ | $10^{-4}$ sec | $10^{-7}$ sec |
| $10^6$ | 5 min | $10^{-4}$ sec |
| $10^9$ | 8 yrs | .28 sec |
| $10^{12}$ | $10^6$ yrs | 4.6 min |

# We are working on it.



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- We made a Finite Difference Mesh based method, which is a variation of the Particle-Particle Particle-Mesh method, that improved efficiency from $\mathcal{O}(n^2)$ to approximately $\mathcal{O}(n)$

# We are working on it.



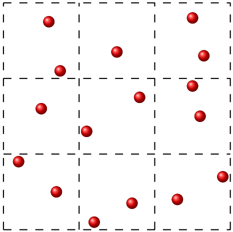
Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

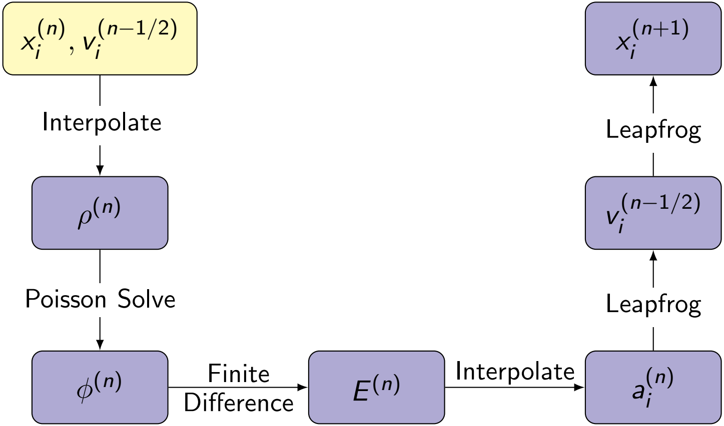
- We made a Finite Difference Mesh based method, which is a variation of the Particle-Particle Particle-Mesh method, that improved efficiency from $\mathcal{O}(n^2)$ to approximately $\mathcal{O}(n)$
- We want to compute the field on the particles quickly

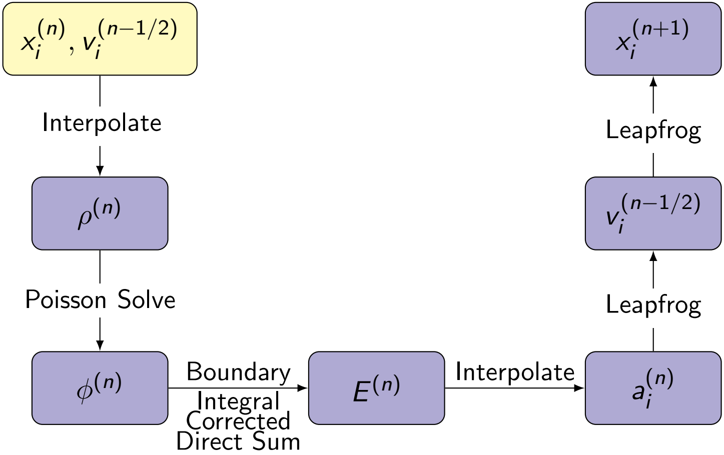
# We are working on it.



- We made a Finite Difference Mesh based method, which is a variation of the Particle-Particle Particle-Mesh method, that improved efficiency from $\mathcal{O}(n^2)$ to approximately $\mathcal{O}(n)$
- We want to compute the field on the particles quickly

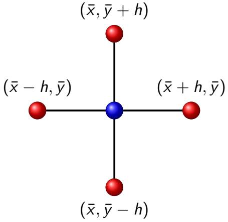



$x_i^{(n)}, v_i^{(n-1/2)}$
Interpolate
$\rho^{(n)}$
Poisson Solve
$\phi^{(n)}$
Finite Difference
$E^{(n)}$
Interpolate
$a_i^{(n)}$
Leapfrog
$v_i^{(n-1/2)}$
Leapfrog
$x_i^{(n+1)}$

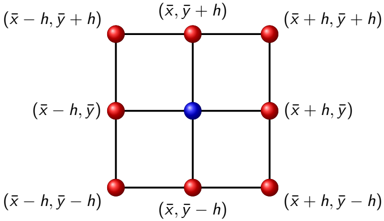

$x_i^{(n)}, v_i^{(n-1/2)}$
Interpolate
$\rho^{(n)}$
Poisson Solve
$\phi^{(n)}$
Boundary Integral Corrected Direct Sum
$E^{(n)}$
Interpolate
$a_i^{(n)}$
Leapfrog
$v_i^{(n-1/2)}$
Leapfrog
$x_i^{(n+1)}$

# Using Finite Difference to Solve



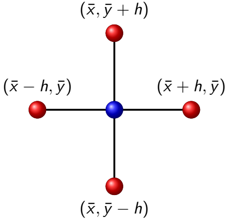



**Five-Point Stencil**

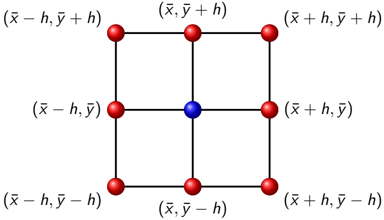



**Nine-Point Stencil**

# Using Finite Difference to Solve



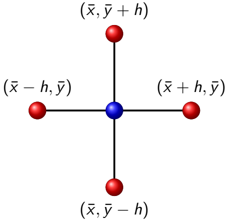

**Five-Point Stencil**



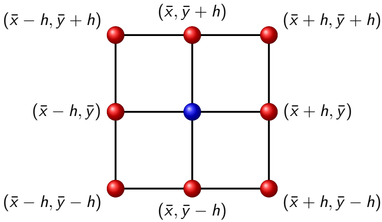

**Nine-Point Stencil**

- $\nabla^2\phi = \partial_{xx}\phi + \partial_{yy}\phi$

# Using Finite Difference to Solve



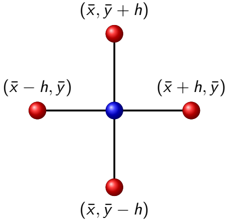

**Five-Point Stencil**



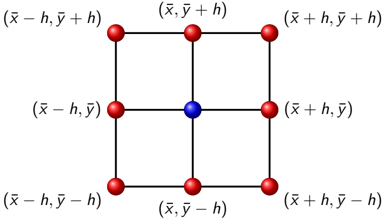

**Nine-Point Stencil**

- $\nabla^2\phi = \partial_{xx}\phi + \partial_{yy}\phi$
- $\nabla^2\phi = \frac{\phi^{j+1,k} - 2\phi^{j,k} + \phi^{j-1,k}}{(\Delta x)^2} + \frac{\phi^{j,k+1} - 2\phi^{j,k} + \phi^{j,k-1}}{(\Delta y)^2} + \mathcal{O}(\Delta x^2 + \Delta y^2)$
  where $j$ is the index for $x$ and $k$ is the index for $y$

# Using Finite Difference to Solve




**Five-Point Stencil**

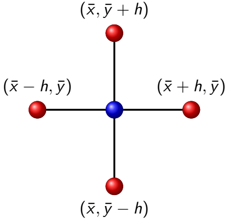



**Nine-Point Stencil**

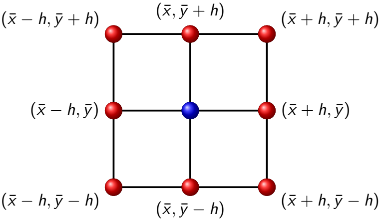
- $\nabla^2\phi = \partial_{xx}\phi + \partial_{yy}\phi$
- $\nabla^2\phi = \frac{\phi^{j+1,k} - 2\phi^{j,k} + \phi^{j-1,k}}{(\Delta x)^2} + \frac{\phi^{j,k+1} - 2\phi^{j,k} + \phi^{j,k-1}}{(\Delta y)^2} + \mathcal{O}(\Delta x^2 + \Delta y^2)$
  where $j$ is the index for $x$ and $k$ is the index for $y$
- $A\vec{\phi} = \vec{\rho}$

# Using Finite Difference to Solve






**Five-Point Stencil**




**Nine-Point Stencil**

- $\nabla^2 \phi = \partial_{xx}\phi + \partial_{yy}\phi$
- $\nabla^2 \phi = \frac{\phi^{j+1,k} - 2\phi^{j,k} + \phi^{j-1,k}}{(\Delta x)^2} + \frac{\phi^{j,k+1} - 2\phi^{j,k} + \phi^{j,k-1}}{(\Delta y)^2} + \mathcal{O}(\Delta x^2 + \Delta y^2)$
  where $j$ is the index for $x$ and $k$ is the index for $y$
- $A\vec{\phi} = \vec{\rho}$
- $E_y(x, y) = \frac{\phi(x,y+h) - \phi(x,y-h)}{2h}$,
  $E_x(x, y) = \frac{\phi(x+h,y) - \phi(x-h,y)}{2h}$

- Acceleration is used to find the velocity at $t = n + .5$

- Acceleration is used to find the velocity at $t = n + .5$
- That velocity is then used to find the position of each particle at $t = n + 1$

- Acceleration is used to find the velocity at $t = n + .5$
- That velocity is then used to find the position of each particle at $t = n + 1$
- This is the leapfrog algorithm

- Acceleration is used to find the velocity at $t = n + .5$
- That velocity is then used to find the position of each particle at $t = n + 1$
- This is the leapfrog algorithm
- $v_{1/2} = v_0 + a_0 \cdot \frac{dt}{2}$
- $v_{n+1/2} = v_{n-1/2} + a_n \cdot dt$
- $x_n = x_{n-1} + v_{n-1/2} \cdot dt$

## Deriving Green's Identity

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} \left(G(x - x_0)\nabla\phi - \phi\nabla G(x - x_0)\right)\cdot\vec{n}ds$$

## Deriving Green's Identity

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G(x - x_0)\nabla\phi - \phi\nabla G(x - x_0)) \cdot \vec{n} ds$$

$$\int_V \nabla \cdot \vec{F} dv = \int_{\partial V} \vec{F} \cdot \vec{n} ds$$

Deriving Green's Identity

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} \left(G(x - x_0)\nabla\phi - \phi\nabla G(x - x_0)\right) \cdot \vec{n} ds$$

$$\int_V \nabla \cdot \vec{F} dv = \int_{\partial V} \vec{F} \cdot \vec{n} ds$$

Let $\vec{F} = \theta\gamma$ where $\theta$ is a scalar function, $\gamma = \nabla\psi$ is a vector function.

$$\int_V \left(\theta\nabla^2\psi + \nabla\theta \cdot \nabla\psi\right) dV = \int_{\partial V} \left(\theta\nabla\psi \cdot \vec{n}\right) ds$$

Deriving Green's Identity

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G(x - x_0)\nabla\phi - \phi\nabla G(x - x_0)) \cdot \vec{n} ds$$

$$\int_V \nabla \cdot \vec{F} dv = \int_{\partial V} \vec{F} \cdot \vec{n} ds$$

Let $\vec{F} = \theta\gamma$ where $\theta$ is a scalar function, $\gamma = \nabla\psi$ is a vector function.

$$\int_V \left(\theta\nabla^2\psi + \nabla\theta \cdot \nabla\psi\right) dV = \int_{\partial V} \left(\theta\nabla\psi \cdot \vec{n}\right) ds$$

If we let $\theta = G(x - x_0)$ and $\psi = \phi$, then we can solve for the potential $\phi(x)$

Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- We can use the fact that $\rho = \sum_{i=1}^{N} \delta(x - x_i)$

- We can use the fact that $\rho = \sum_{i=1}^{N} \delta(x - x_i)$
- We can solve for $\phi$ to obtain:

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G \nabla \phi - \phi \nabla G) \cdot \vec{n} ds$$

- We can use the fact that $\rho = \sum_{i=1}^{N} \delta(x - x_i)$
- We can solve for $\phi$ to obtain:

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G\nabla\phi - \phi\nabla G) \cdot \vec{n} ds$$

- $\phi_{x,j} = \sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{-1}{2\pi} \ln\left(||x_i - x_j||_2\right) + \oint_{\partial\Omega} (\phi\nabla G - G\nabla\phi) \cdot \vec{n}\, ds$

- We can use the fact that $\rho = \sum_{i=1}^{N} \delta(x - x_i)$
- We can solve for $\phi$ to obtain:

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G\nabla\phi - \phi\nabla G) \cdot \vec{n} ds$$

- $\phi_{x,j} = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{-1}{2\pi} \ln\left(||x_i - x_j||_2\right) + \oint_{\partial\Omega} (\phi\nabla G - G\nabla\phi) \cdot \vec{n}\, ds$
- New equation because Potential Theory tells us it is the same $\phi_{x,j} = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{-1}{2\pi} \ln\left(||x_i - x_j||_2\right) + \oint_{\partial\Omega} \sigma_s(y) G\left(x_j - y\right)\, ds(y)$

- We can use the fact that $\rho = \sum_{i=1}^{N} \delta(x - x_i)$
- We can solve for $\phi$ to obtain:

$$\phi(x) = \sum_{i=1}^{N} G(x_i - x_0) - \int_{\partial V} (G\nabla\phi - \phi\nabla G) \cdot \vec{n} ds$$

- $\phi_{x,j} = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{-1}{2\pi} \ln\left(||x_i - x_j||_2\right) + \oint_{\partial\Omega} (\phi\nabla G - G\nabla\phi) \cdot \vec{n}\, ds$
- New equation because Potential Theory tells us it is the same $\phi_{x,j} = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{-1}{2\pi} \ln\left(||x_i - x_j||_2\right) + \oint_{\partial\Omega} \sigma_s(y) G\left(x_j - y\right)\, ds(y)$
- $\sigma$ is like a surface charge

- We use direct sum to calculate the field due to local particles

- We use direct sum to calculate the field due to local particles
- In order to evaluate the boundary integral, we solve using numerical quadrature.

- We use direct sum to calculate the field due to local particles
- In order to evaluate the boundary integral, we solve using numerical quadrature.
- We end up with a matrix that we have to invert to obtain the surface charge.

- We use direct sum to calculate the field due to local particles
- In order to evaluate the boundary integral, we solve using numerical quadrature.
- We end up with a matrix that we have to invert to obtain the surface charge.
- Then, instead of doing this everywhere which would be $\mathcal{O}(n^2)$, we make this a subcell method which gives $\mathcal{O}(n) + \mathcal{O}(k^2)$.

# Solving for Sigma



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- $\int_{s_1}^{s_2} \sigma G \, ds$

# Solving for Sigma



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- $\int_{s_1}^{s_2} \sigma G \, ds$
- We take a limit to the boundary with respect to variable $x$ and get the integral equation

# Solving for Sigma



- $\int_{s_1}^{s_2} \sigma G \, ds$
- We take a limit to the boundary with respect to variable $x$ and get the integral equation
- We turn the integral into discrete approximations that we can solve

# Solving for Sigma



- $\int_{s_1}^{s_2} \sigma G \, ds$
- We take a limit to the boundary with respect to variable $x$ and get the integral equation
- We turn the integral into discrete approximations that we can solve
- The integral becomes four sums

# Solving for Sigma



Ben Lewis, Bridget Morales, David Marsico, Jason McKelvey, Madie Wilkin

- $\int_{s_1}^{s_2} \sigma G \, ds$
- We take a limit to the boundary with respect to variable $x$ and get the integral equation
- We turn the integral into discrete approximations that we can solve
- The integral becomes four sums
- We use integral mean value theorem and trapezoid rule to end up with the Green's matrix
- $\sum_{i=1}^{4k} \int_{s_1}^{s_2} \sigma G \, ds$

# Solving for Sigma

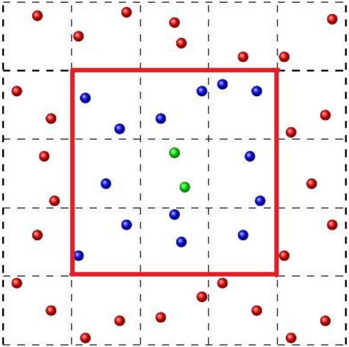


- $\int_{s_1}^{s_2} \sigma G \, ds$
- We take a limit to the boundary with respect to variable $x$ and get the integral equation
- We turn the integral into discrete approximations that we can solve
- The integral becomes four sums
- We use integral mean value theorem and trapezoid rule to end up with the Green's matrix
- $\sum_{i=1}^{4k} \int_{s_1}^{s_2} \sigma G \, ds$
- Then we end up with a system of equations in which we need to perform a Green's matrix inversion

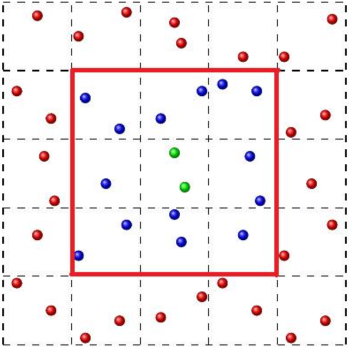
In order to get the Boundary Integral Corrected Direct Sum Electric Fields we need several things

In order to get the Boundary Integral Corrected Direct Sum Electric Fields we need several things

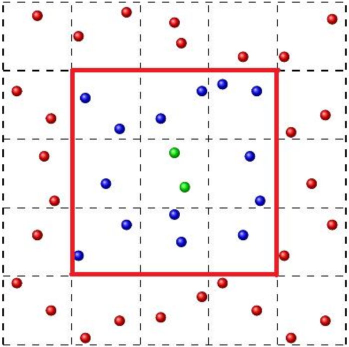
- List of all particles inside the cell



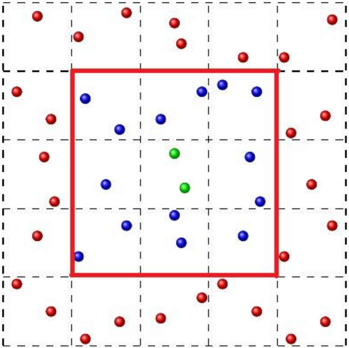
- List of all particles in boundary cells



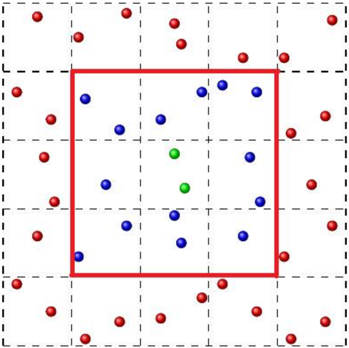
- Potential at all 12 border points per cell



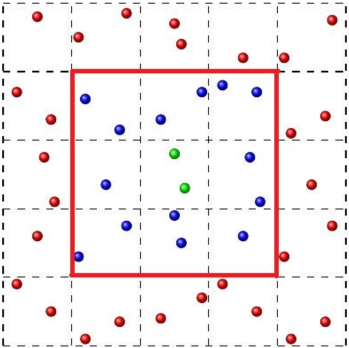
- Something to take the above and calculate the electric field



- For loop over all cells



- For loop over all cells



If we can get this done, then it's a matter of increasing customizability

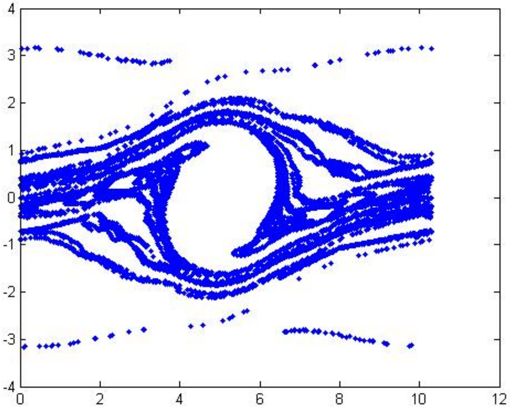

4
3
2
1
0
-1
-2
-3
-4
0
2
4
6
8
10
12

Figure: Uncorrected

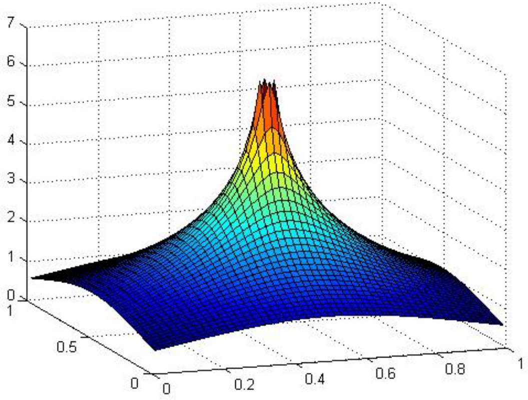


Figure: Point charge at the center