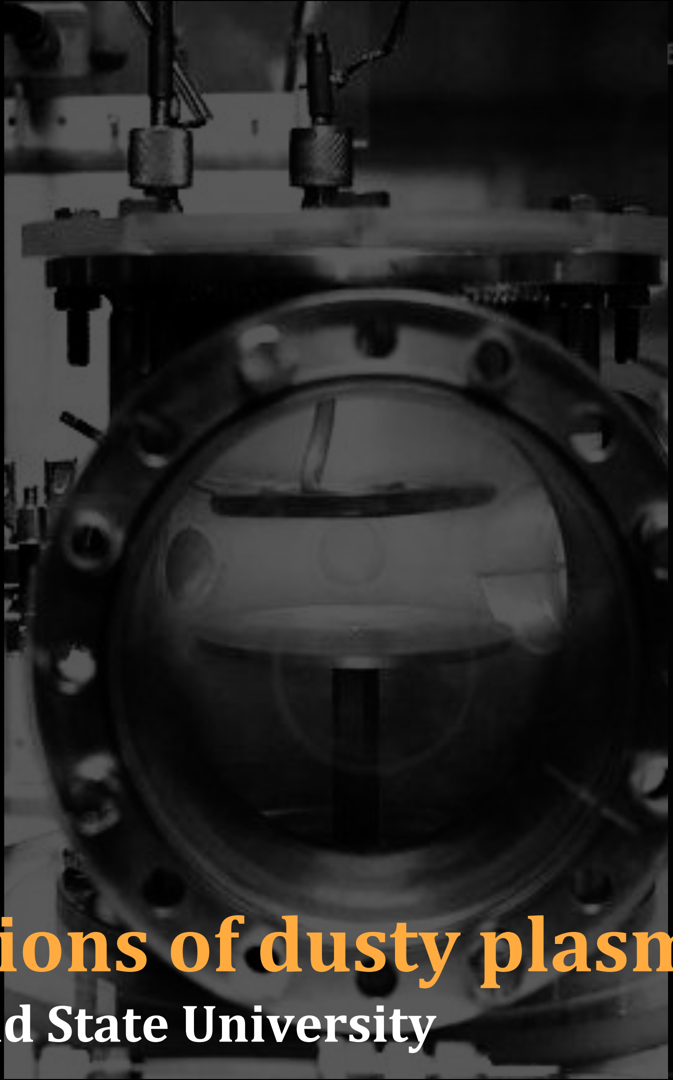


Hybrid simulations of dusty plasma

I. J. Rodriguez | Portland State University

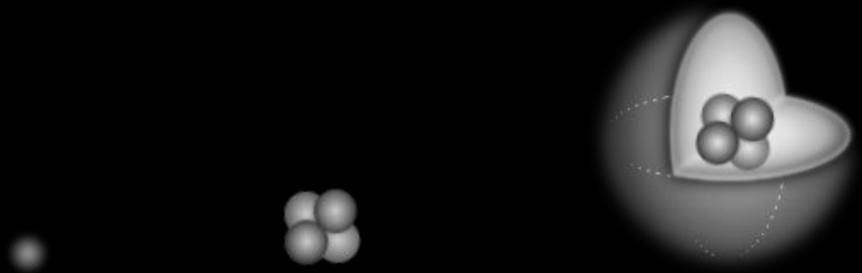

High-altitude clouds

Saturn’s rings

Rocket exhaust

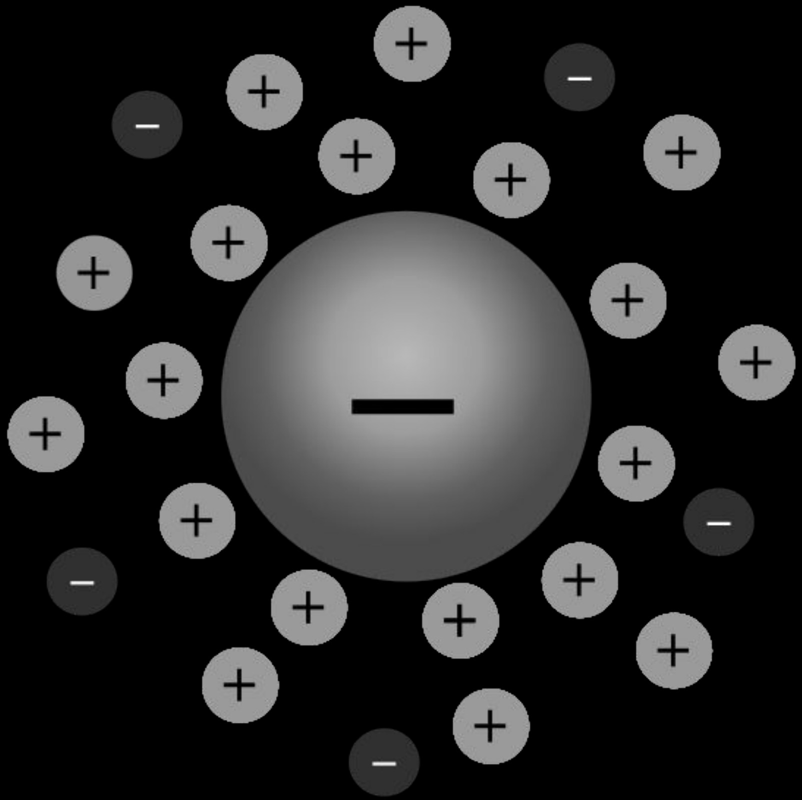
PDF version. Click below to play video.

Valderrama, E. (2008). University of California San Diego electrons + ions + neutrals + dust = *dusty* plasma



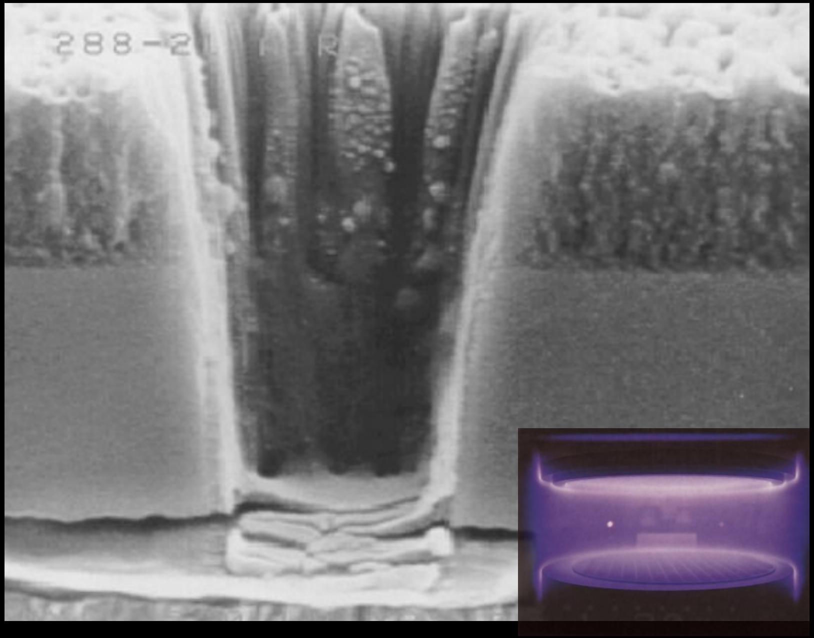
Dust tends to:

- acquire a negative charge
- cluster and grow
- accumulate at boundaries

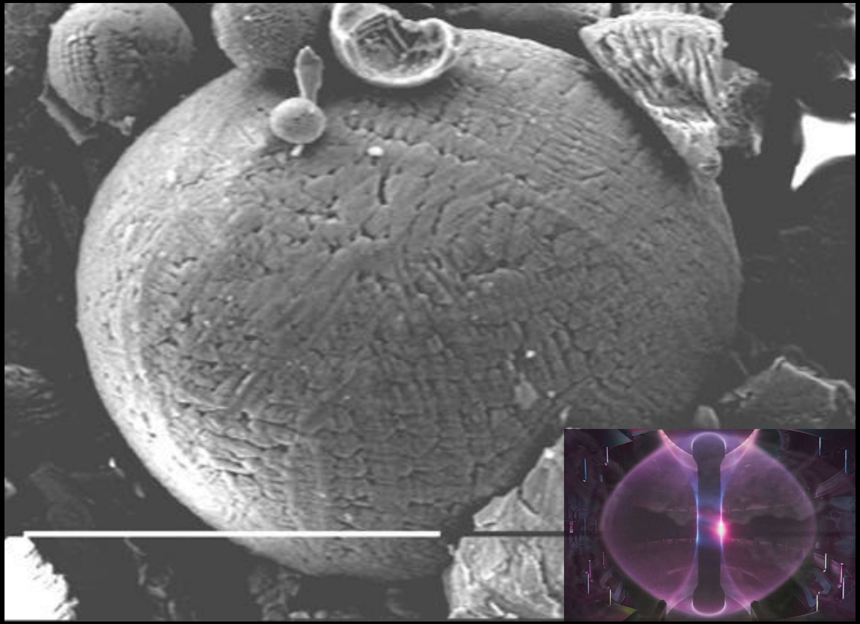


The presence of dust in plasmas can be problematic.

Layers of contamination



Ejected debris



$$\vec{\nabla} \cdot \vec{B} = 0$$

$$\vec{\nabla} \times \vec{E} + \frac{1}{c}\frac{\partial B}{\partial t} = 0$$

$$\vec{\nabla} \cdot \vec{E} = 4\pi\rho$$

Theoretical

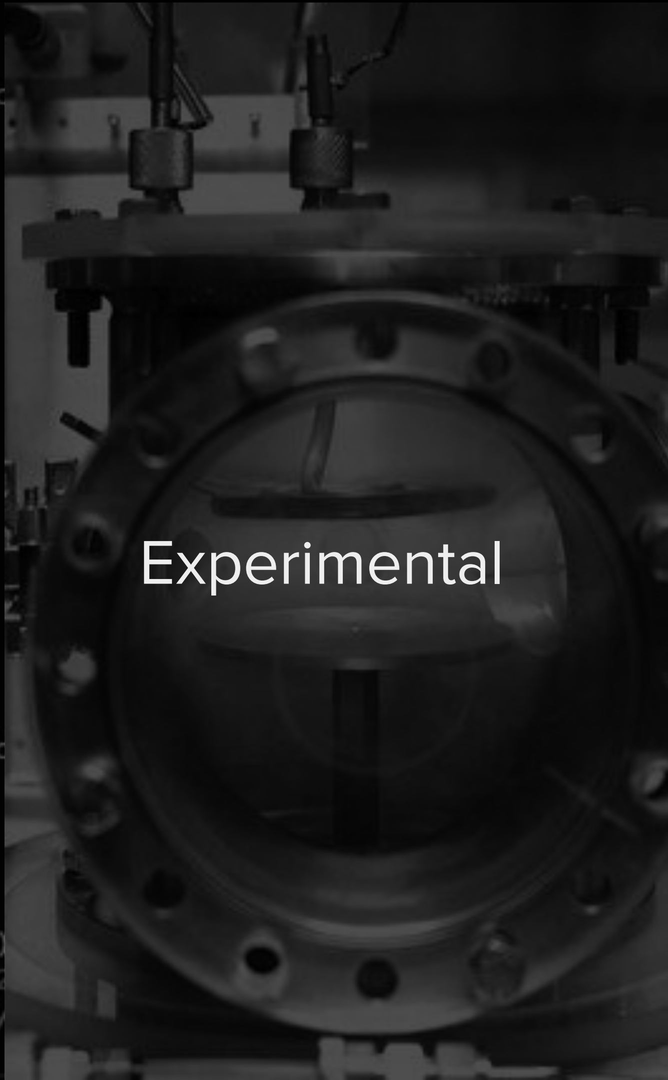
$$\vec{\nabla} \times \vec{B} - \frac{1}{c}\frac{\partial E}{\partial t} = \frac{4\pi}{c}\vec{J}$$

$$\vec{B} = \vec{\nabla} \times \vec{A}$$

$$\vec{E} = -\vec{\nabla}\phi - \frac{1}{c}\frac{\partial A}{\partial t}$$

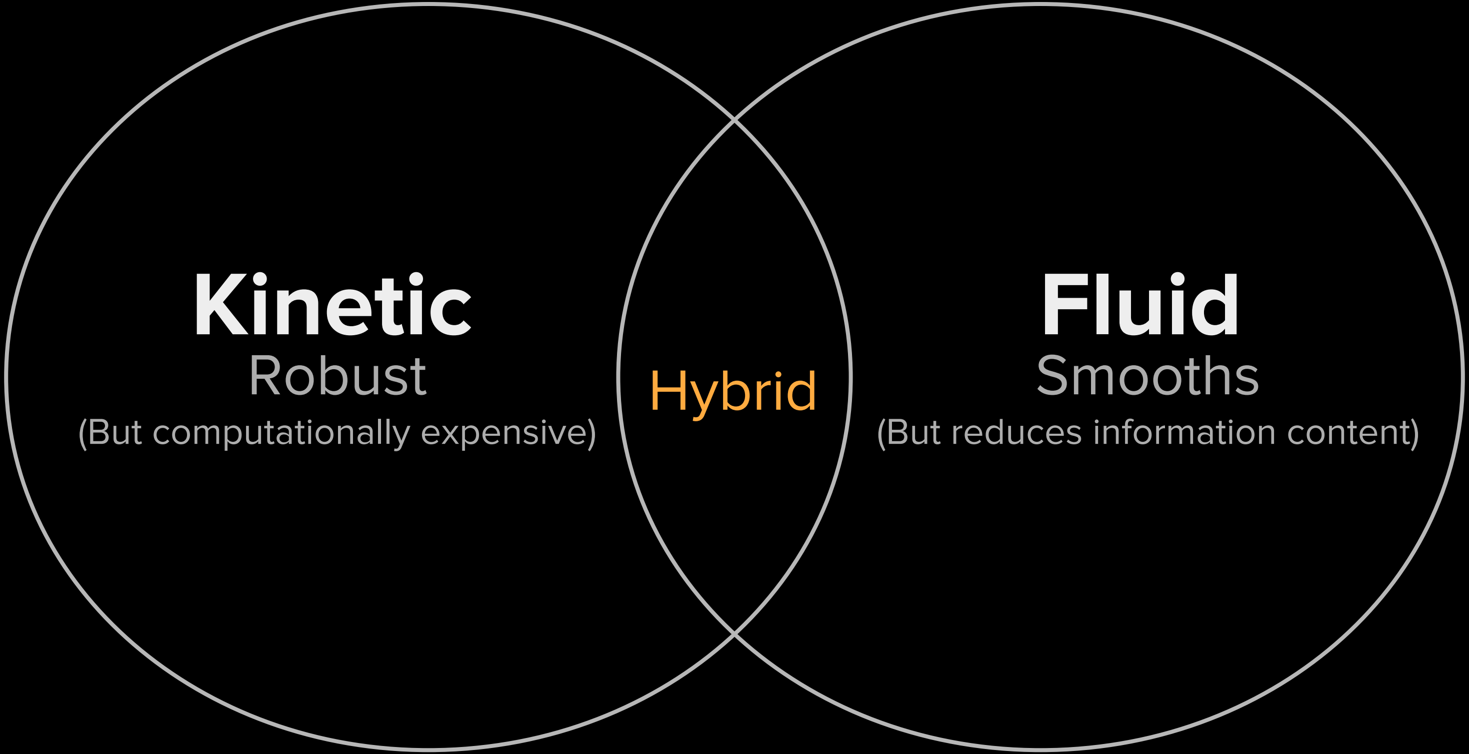
```
E_from_V(rho, J, dx):
"""Uses the finite differenc

source = rho[0:J-1]*dx**2
M = np.zeros((J-1,J-1))

for i in range(0, J-1):
    for j in range(0, J-1):
        if i == j:
            M[i,j] = 2.
        if i == j-1:
            M[i,j] = -1.
        if i == j+1:

M[0, J-2] = -1.
M[J-2, 0] = -1.

V = np.linalg.solve(M, sourc

E = np.zeros((J,1))

for i in range (1,J-2):
    E[i] = (V[i+1] - V[i-1])
E[J-2] = (V[0] - V[J-3]) / 2
E[0] = (V[1] - V[J-2]) / 2./
E[J-1] = E[0]
```

Computational



Experimental

Plasma simulations can be divided into two main categories.



Goal: Create a hybrid simulation using Python that can replicate experimental results.

Kinetic

- Electrons
  - Electrostatic Particle In Cell
  - *Monte-Carlo Collisions*

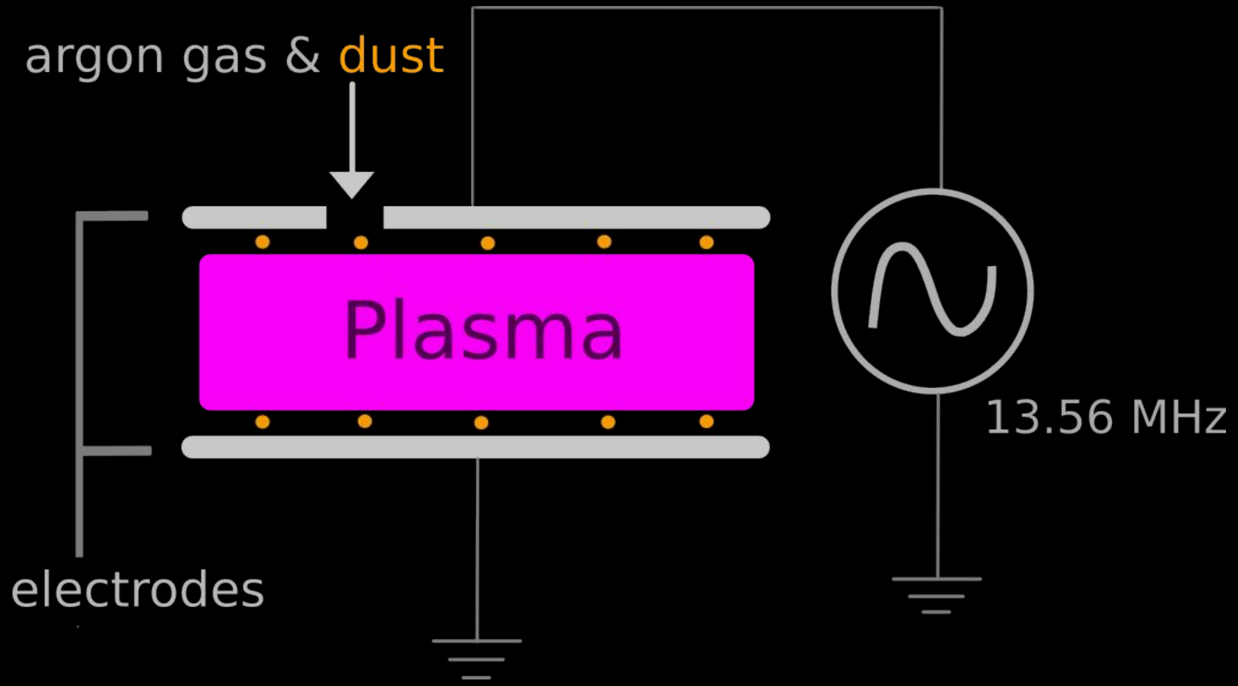
Fluid

- *Dust*
- *Ions*

```
E_from_V(rho, J, dx):
"""Uses the finite differe

source = rho[0:J-1]*dx**2
M = np.zeros((J-1,J-1))

for i in range(0, J-1):
    for j in range(0, J-1)
        if i == j:
            M[i,j] = 2.
        if i == j-1:
            M[i,j] = -1.
        if i == j+1:
            M[i,j] = -1.

M[0, J-2] = -1.
M[J-2, 0] = -1.

V = np.linalg.solve(M, sou

E = np.zeros((J,1))

for i in range (1,J-2):
    E[i] = (V[i+1] - V[i-1
E[J-2] = (V[0] - V[J-3]) /
E[0] = (V[1] - V[J-2]) / 2
E[J-1] = E[0]
```

Pustynik et al. (2017)

CCRF Discharge Plasma



ES PIC code tested against the benchmark two-stream instability problem.

PDF version. Click below to play video.

What’s next?

Kinetic
- *Electrons*
  - *ES Particle-in-cell*
  - Monte-Carlo Collisions

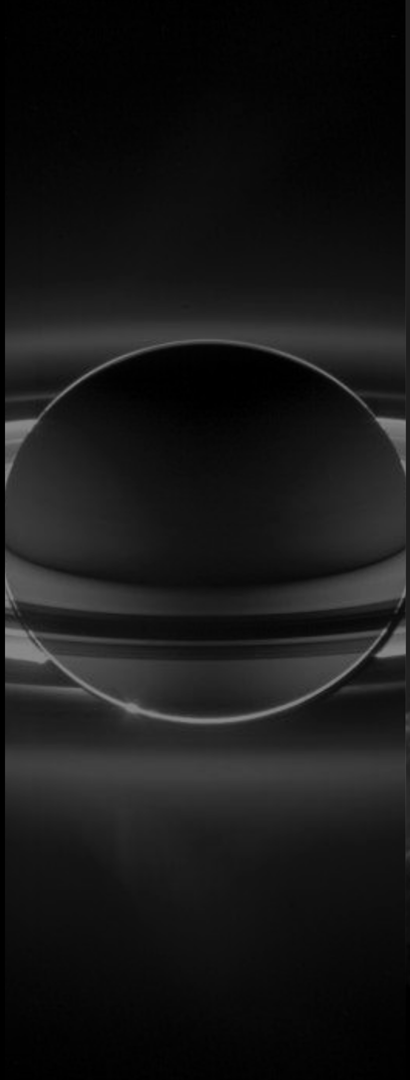
Fluid
- Dust
- Ions

```
E_from_V(rho, J, dx):
"""Uses the finite differe

source = rho[0:J-1]*dx**2
M = np.zeros((J-1,J-1))

for i in range(0, J-1):
    for j in range(0, J-1)
        if i == j:
            M[i,j] = 2.
        if i == j-1:
            M[i,j] = -1.
        if i == j+1:
            M[i,j] = -1.

M[0, J-2] = -1.
M[J-2, 0] = -1.

V = np.linalg.solve(M, sou

E = np.zeros((J,1))

for i in range (1,J-2):
    E[i] = (V[i+1] - V[i-1
E[J-2] = (V[0] - V[J-3]) /
E[0] = (V[1] - V[J-2]) / 2
E[J-1] = E[0]
```

Pustynik et al., 2017

Wrapping up

- Dusty plasma is everywhere!
- Simulations can help fill in the gaps



github.com/space-isa