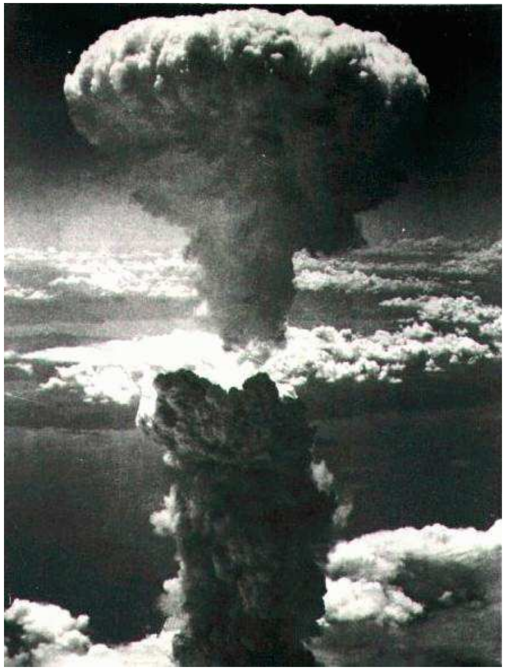
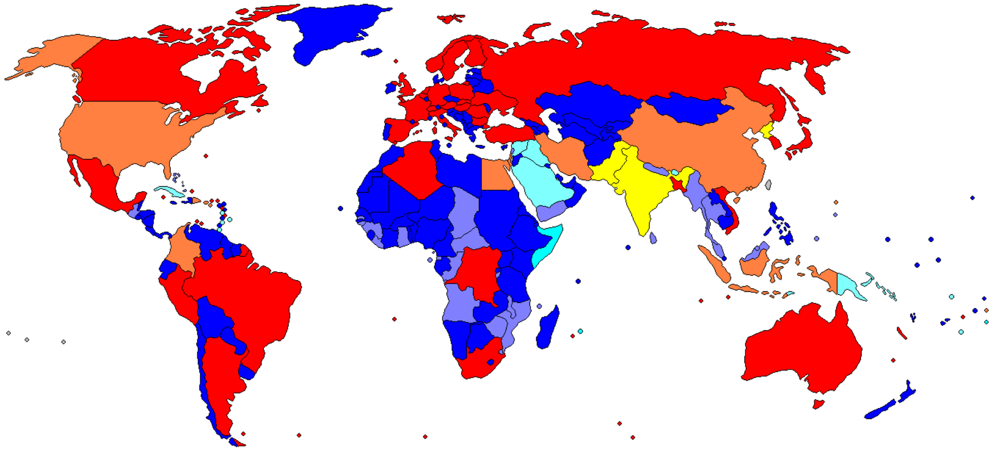
The Comprehensive Test Ban Treaty (CTBT)

- The CTBT bans all nuclear explosions for military or civilian purposes to limit the proliferation of nuclear weapons by cutting a vital link, testing, in their development.
- A network of seismological, hydroacoustic, infrasound, and radionuclide sensors will monitor compliance. Once the Treaty enters into force, on-site inspection will be provided to check compliance.
- The US has signed the CTBT, but not ratified it.





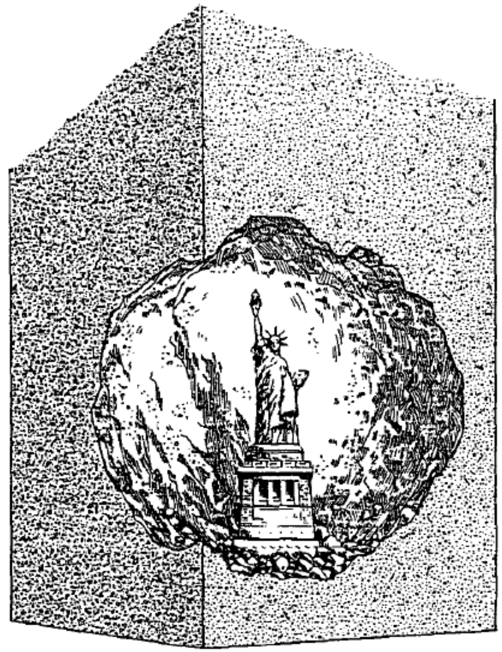
Red, Blue - ratified
Orange, Azure - signed
Yellow, Cyan - outside treaty

Can an Opponent Cheat on the CTBT?

- U.S. and Russian experiments have demonstrated that seismic signals can be muffled, or decoupled, for a nuclear explosion detonated in a large underground cavity.
- Such technical scenarios are credible only for yields of at most a few kilotons.
- Seismic component of the International Monitoring Syster consist of 170 seismic stations.
- The INS is expected to detect all seismic events of about magnitude 4 or larger corresponds to an explosive yield of approximately 1 kiloton (the explosive yield of 1,000 tons of TNT).

What can be learned from low-yield, surreptitious blasts?

Can it extrapolated to full-up tests?



Demonstration of size of cavity needed to decouple a 5 kT blast.

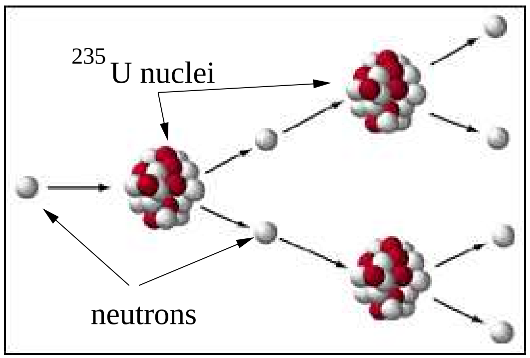
Nuclear Weapons 101

- Fissile materials (^{235}U, ^{233}U, ^{239}Pu) are used to make weapons of devastating power.
- As each nucleus fissions, it emits 2 or so neutrons plus lots of energy. Usually most of the neutrons leave without striking any other nuclei.

$$^{235}\mathrm{U} + \mathrm{n} \rightarrow {}^{236}\ \mathrm{U}^{*} \rightarrow {}^{140}\ \mathrm{Xe}\ +\ {}^{94}\mathrm{Sr}\ +\ 2\mathrm{n}\ +\ \approx 200\ \mathrm{MeV}$$

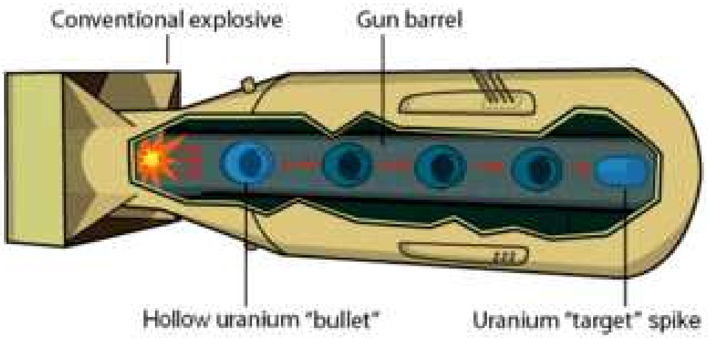
- Increasing the density creates a ‘chain reaction’ where the emitted neutrons cause other fissions in a self-propagating process.
- Only about 8 kg of plutonium or 25 kg of highly-enriched uranium (HEU) is needed is needed to produce a weapon.

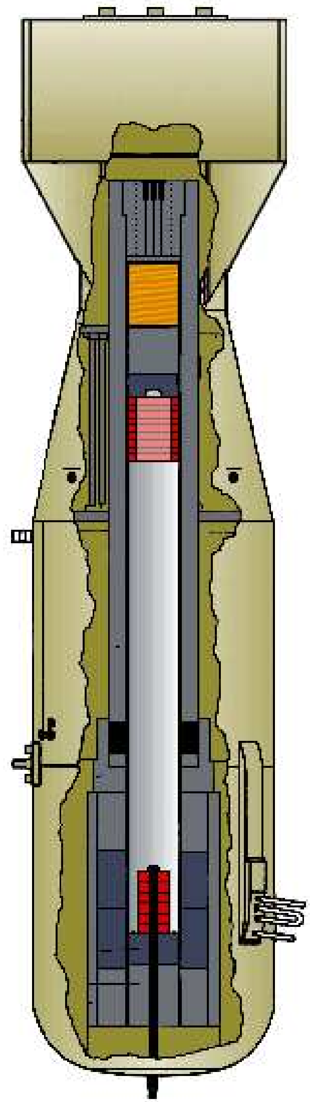
A Chain Reaction



HEU Gun-Type Design

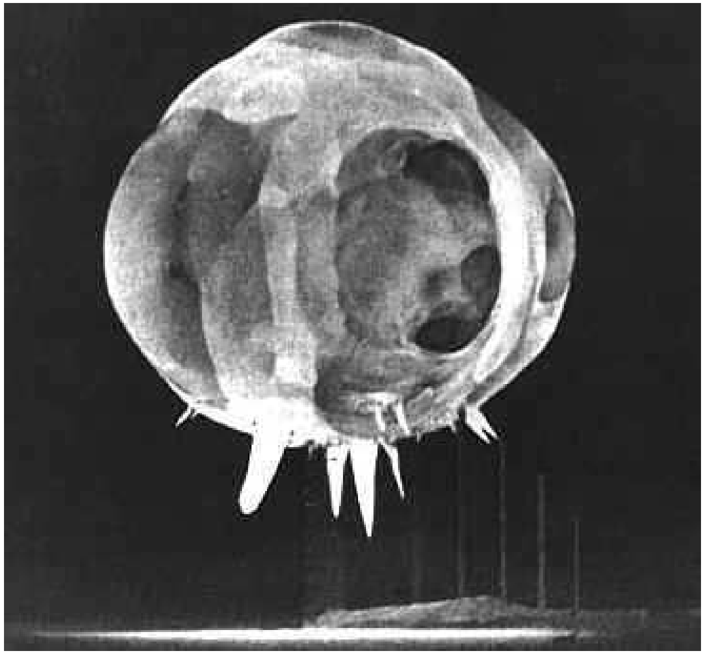
The figure to the right shows the 'Little Boy' design of the nuclear bomb dropped on Hiroshima. The fissile, $^{235}\mathrm{U}$ is shown in red. A cordite charge was detonated behind one of the pieces of $^{235}\mathrm{U}$ accelerating it to a speed of $300\ m/s$ before it struck the target to form a critical mass (see figure below). A neutron trigger/initiator was used to start the chain reaction.





Critical Mass

In the greatest gathering of scientific talent in human history, the Manhattan Project had the goal 'to produce a practical military weapon in the form of a bomb in which the energy is released by a fast neutron chain reaction'. This chain reaction will occur when the neutron number density $n(\vec{r}, t)$ grows exponentially in time. Under what conditions will this occur given the fissile material $^{235}\mathrm{U}$ has a neutron diffusion constant $D = 10^5\ m^2/s$ and a neutron creation rate $C = 10^8\ s^{-1}$?



Nuclear fireball $1\ ms$ after detonation showing rope tricks (Tumbler Snapper).

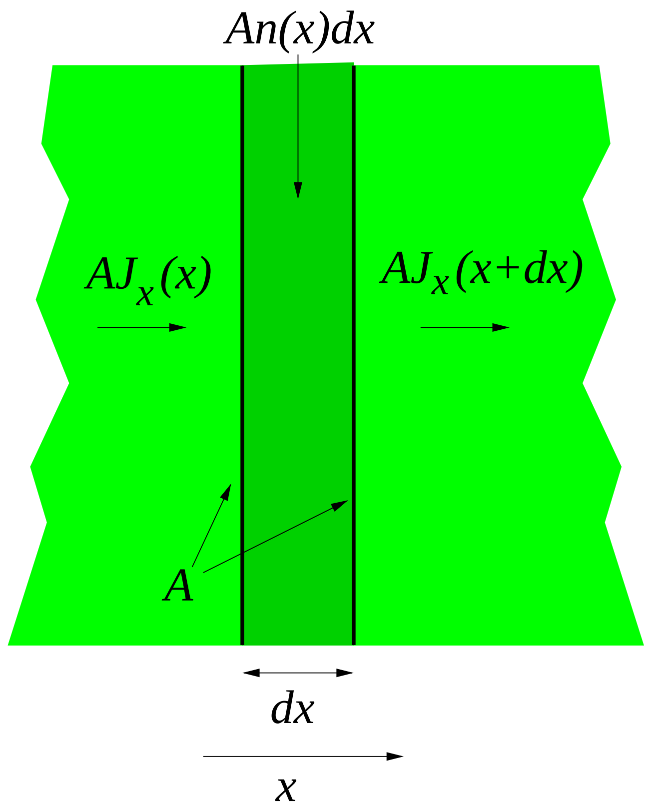
Treat the system as a one-dimensional one of length L in the range $0 < x < L$. Neutrons that reach the boundaries escape and no longer contribute to the reaction so require that $n(x = 0, t) = n(x = L, t) = 0$.

The Diffusion Equation - Getting Started

- Consider a portion of a distribution of matter in a pipe of area A where the number density n depends on position in the x direction.
- Frick's Law describes the flow of material through volume

$$J_x = -D\frac{\partial n}{\partial x}$$

where J_x is the x-component of the flow of material (units: $particles/m^2 - s$), n is the number density of the material, and D is a constant of proportionality (unit:m^2/s).



The Diffusion Equation - An Example

Consider the one-dimensional diffusion equation corresponding to particles in a long pipe of length L.

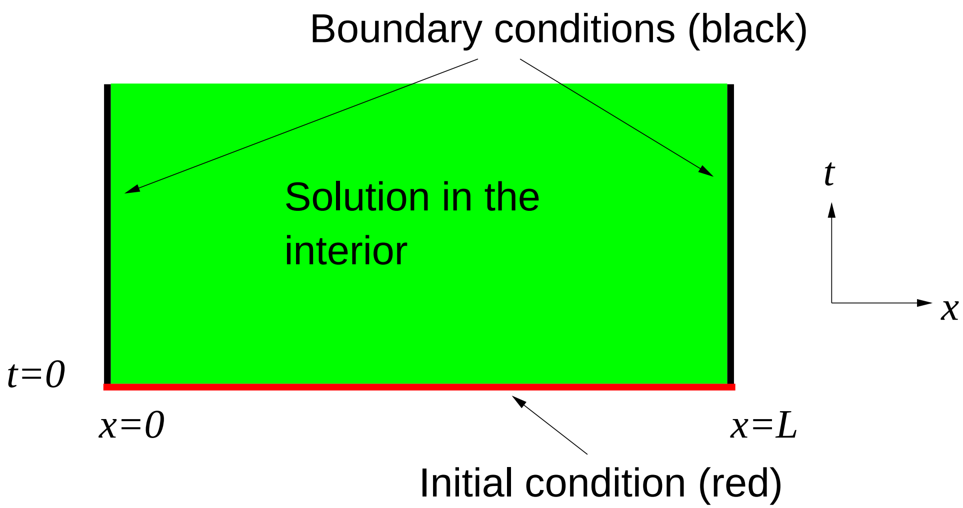
$$\frac{\partial n(x,t)}{\partial t} = D\frac{\partial^2 n}{\partial x^2} + Cn$$

where $n(x,t)$ is the particle density, D is the self-diffusion coefficient, and C is the creation rate. Restrict the problem to the case where there are no sources of particles ($C = 0$).

1. What is the general solution to this differential equation?
2. What restrictions are there on the parameters of the solution?
3. Suppose the particle density goes to zero at the ends of the pipe so $n(x = 0, t) = n(x = L, t) = 0$. What is the particular solution?

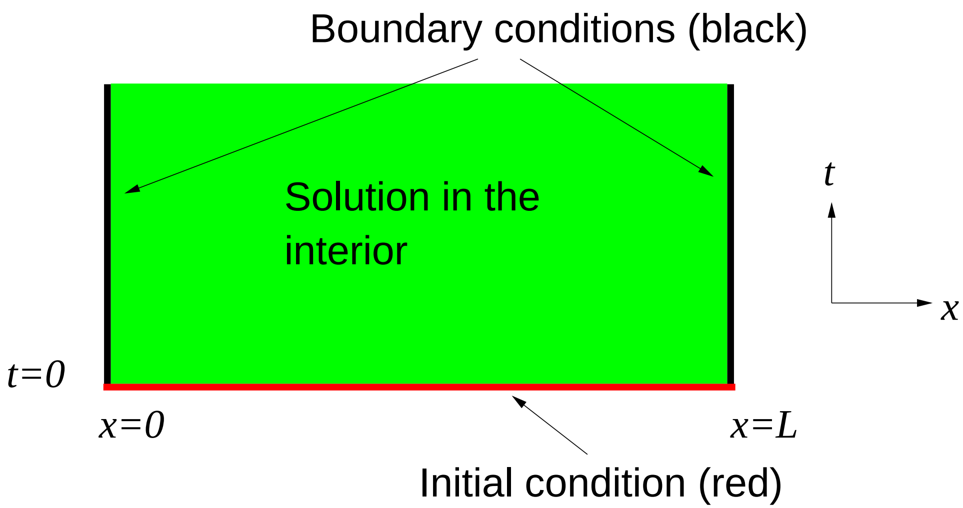
The Diffusion Equation - Discretization

A schematic view of the initial values and boundary conditions.

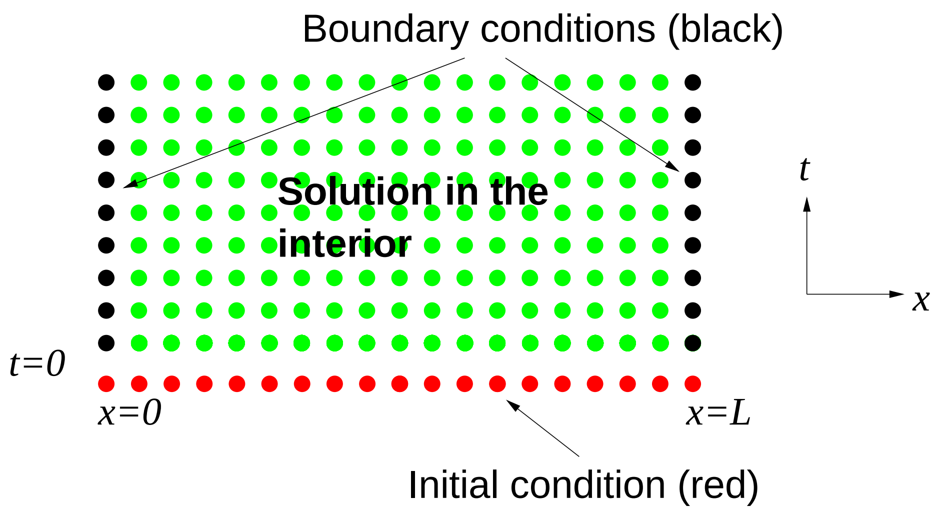


The Diffusion Equation - Discretization

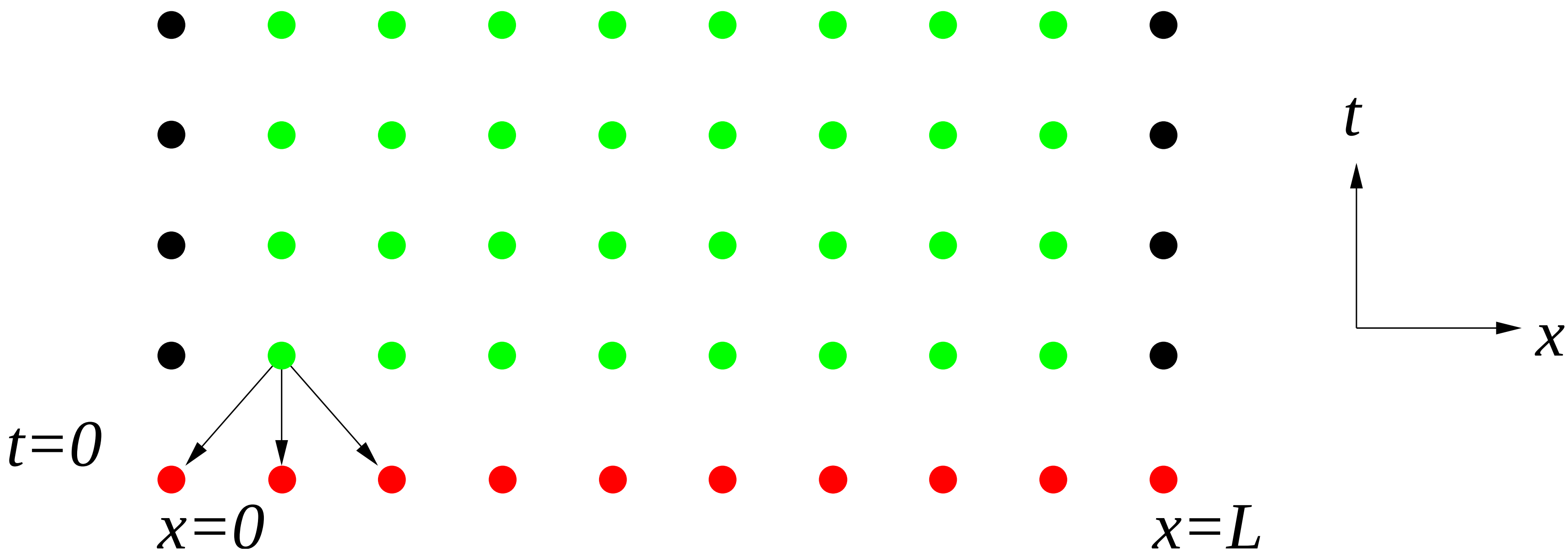
A schematic view of the initial values and boundary conditions.



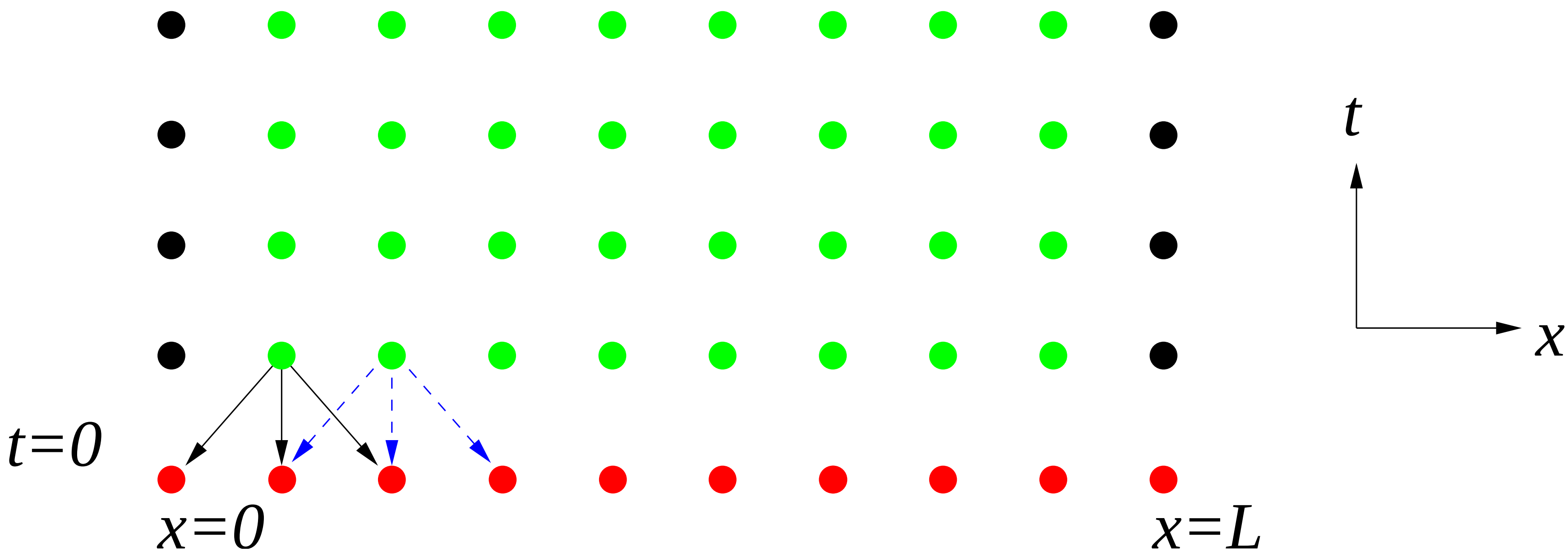
Now discretize the initial values and boundary conditions.



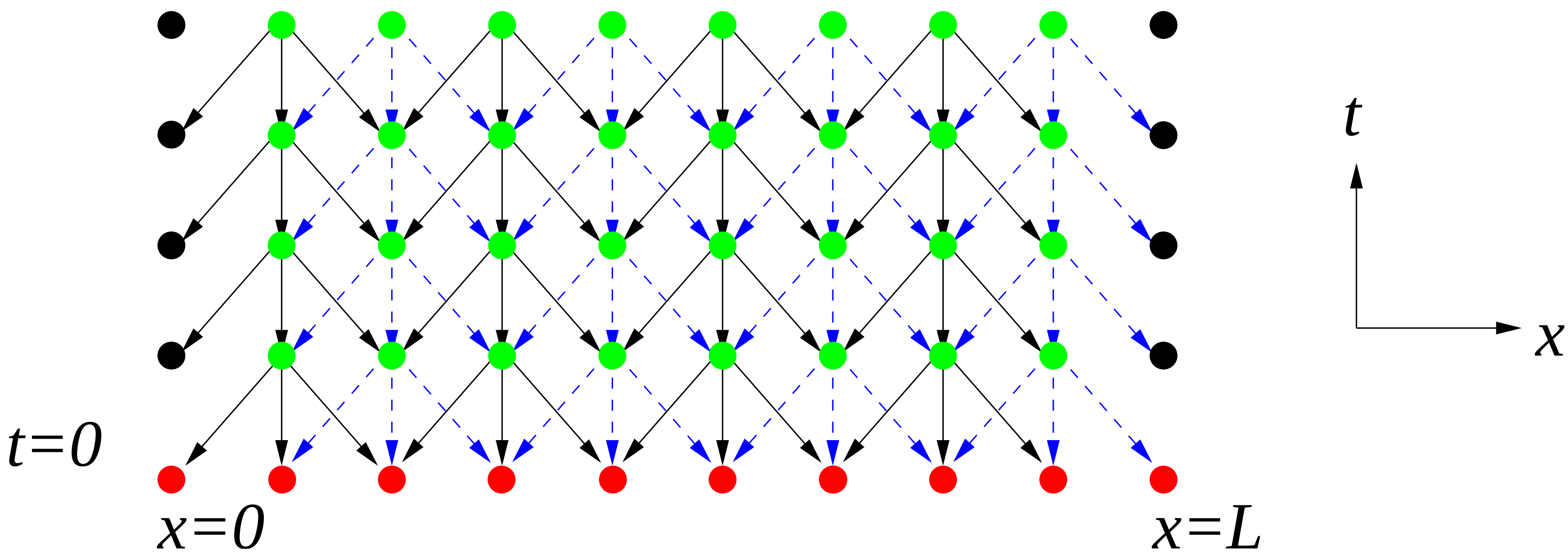
The Diffusion Equation - The Couplings - 1



The Diffusion Equation - The Couplings - 2



The Diffusion Equation - The Couplings - 3



Euler's Relation - 1

Euler's relation (also known as *Euler's formula*) is considered the first bridge between the fields of algebra and geometry, as it relates the exponential function to the trigonometric sine and cosine functions.

Euler's relation states that

$$e^{ix} = \cos x + i \sin x$$

Start by noting that

$$i^k = \begin{cases} 1 & k \equiv 0 \\ i & k \equiv 1 \\ -1 & k \equiv 2 \\ -i & k \equiv 3 \end{cases}$$

Euler's Relation - 2

Using the Taylor series expansions of e^x, $\sin x$ and $\cos x$ it follows that

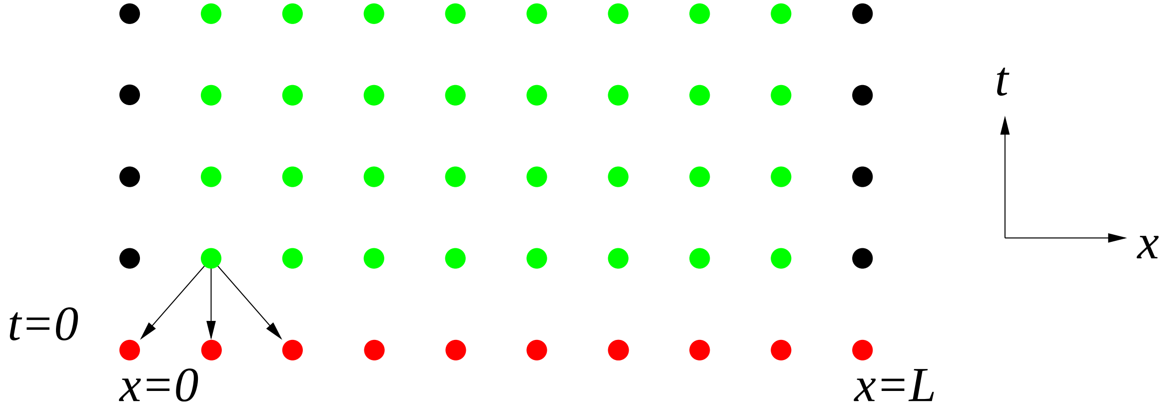
$$e^{ix} = \sum_{n=0}^{\infty} \frac{i^n x^n}{n!} = \sum_{n=0}^{\infty} \left(\frac{x^{4n}}{(4n)!} + \frac{ix^{4n+1}}{(4n+1)!} - \frac{x^{4n+2}}{(4n+2)!} - \frac{ix^{4n+3}}{(4n+3)!} \right)$$

Because the series expansion above is absolutely convergent for all x, we can rearrange the terms of the series as

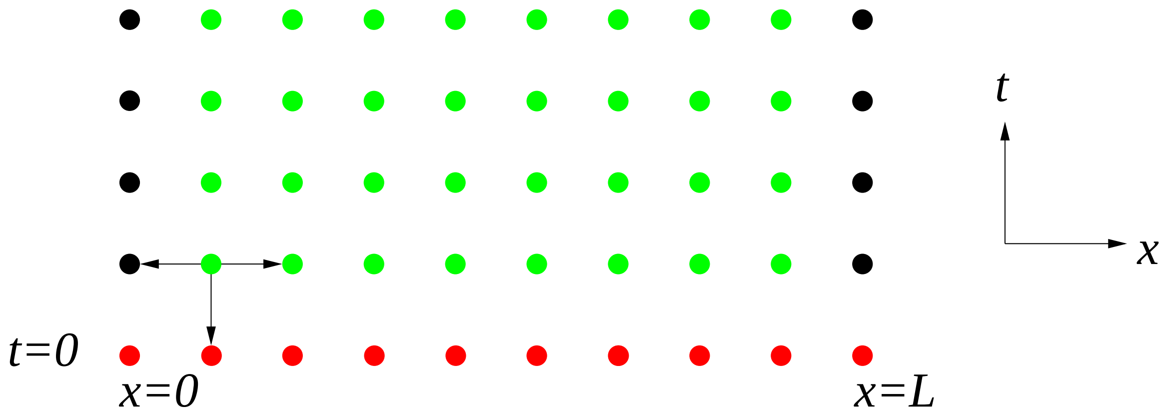
$$e^{ix} = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} + i \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = \cos x + i \sin x$$

The Diffusion Equation - The Couplings - 4

Explicit method.



Implicit method.



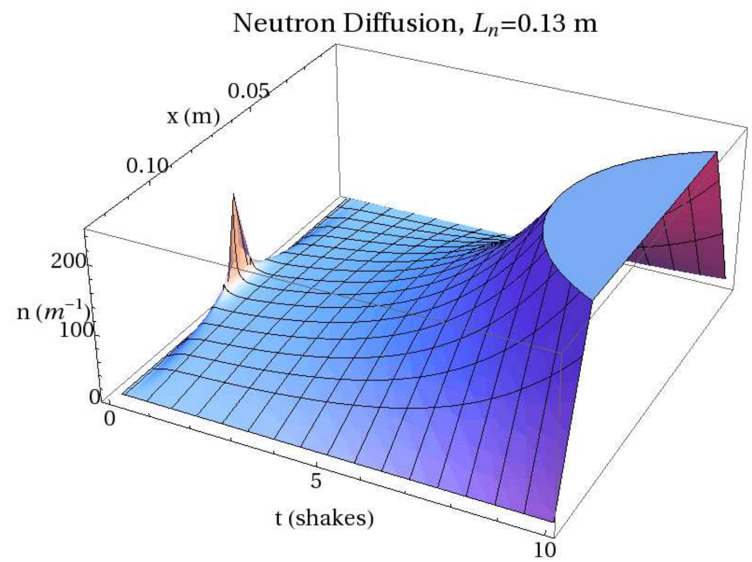
Sample Code - 1

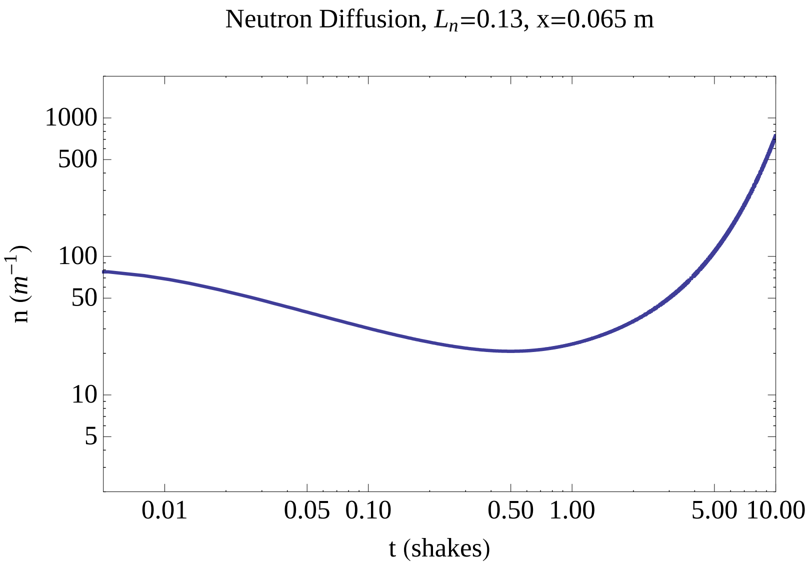
```
(* Define diffusion parameters. *)
Dn = 0.001;  (* self diffusion coefficient in m^2/shake *)
Ln = 0.1;  (* size of the region in meters. *)

(* parameters for the algorithm. *)
tmax = 10.0;  (* maximum time in shakes. *)
Nxsteps = 20;  (* steps in x. *)
Ntsteps = 1000;  (* steps in time. *)
dx = Ln/Nxsteps;  (* stepsize in x (m). *)
dt = tmax/Ntsteps;  (* stepsize in time. *)

(* set up the distribution of particles at t=0 so there is always
a spike of the same size in the middle. *)
n0 = Table[{x, 0, 0}, {x, 0, Ln, dx}];
n0[[Nxsteps/2 + 1, 3]] = 1/dx ;

(* initialize the main array. *)
particle = Table[0.0, {i, 1, Nxsteps}, {n, 1, Ntsteps}];
```

Sample Code - 2

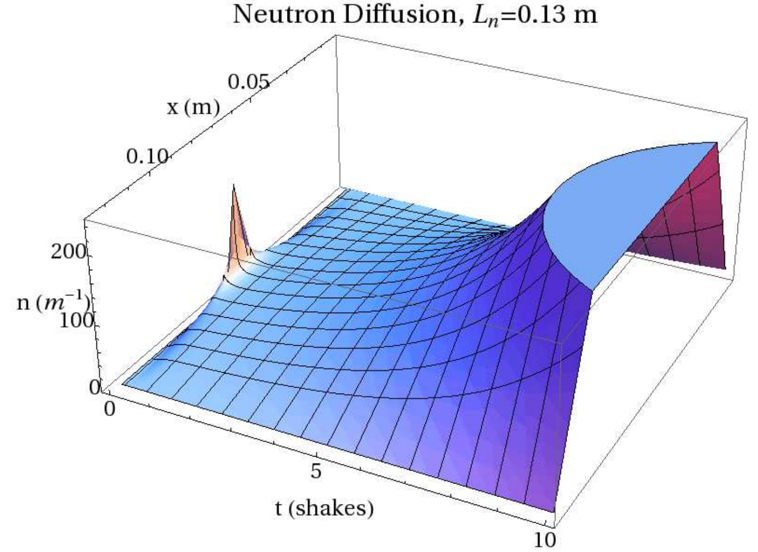
```
(* put in the initial conditions for t=0. *)
Do[particle[[i, 1]] = n0[[i, 3]], {i, 1, Nxsteps}];

(* The boundary condition at x=0. *)
Do[particle[[1, n]] = 10.0, {n, 2, Ntsteps}];
(* The boundary condition at x=L. *)
Do[particle[[Nxsteps, n]] = 0.6, {n, 2, Ntsteps}];

(* constants for the recursion relation. *)
A0 = 1 - (2*dt*Dn)/dx^2;
B0 = (dt*Dn)/dx^2;

(* main loop. outer loop over time and inner loop over position. *)
Do[
 Do[particle[[i, n]] = A0*particle[[i, n - 1]] +
                       B0*particle[[i + 1, n - 1]] +
                       B0*particle[[i - 1, n - 1]],
  {i, 2, Nxsteps - 1}](* end of inner loop *),
 {n, 2, Ntsteps}](* end of outer loop *)
```

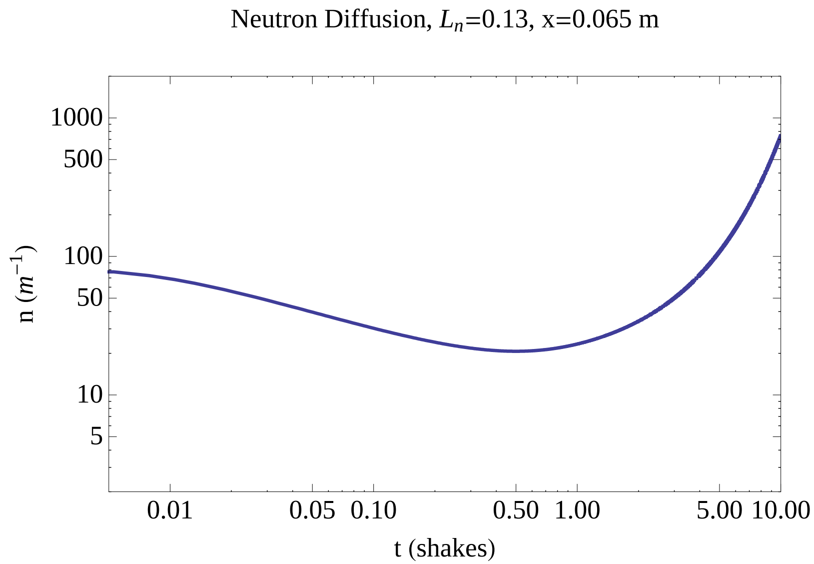
Oh-Oh

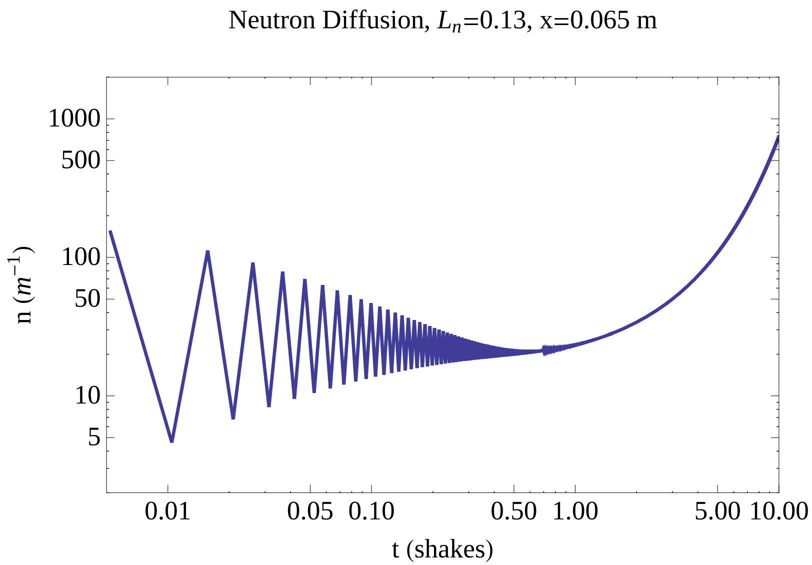

Neutron Diffusion, L_n=0.13 m
x (m)
0.05
0.10
200
n (m^{-1})
100
0
0
5
t (shakes)
10


Neutron Diffusion, L_n=0.13, x=0.065 m
1000
500
100
50
10
5
n (m^{-1})
0.01
0.05
0.10
0.50
1.00
5.00
10.00
t (shakes)

Oh-Oh


Neutron Diffusion, L_n=0.13 m
x (m)
0.05
0.10
200
n (m^{-1})
100
0
0
5
t (shakes)
10


Neutron Diffusion, L_n=0.13, x=0.065 m
1000
500
100
50
10
5
n (m^{-1})
0.01
0.05
0.10
0.50
1.00
5.00
10.00
t (shakes)


Neutron Diffusion, L_n=0.13, x=0.065 m
1000
500
100
50
10
5
n (m^{-1})
0.01
0.05
0.10
0.50
1.00
5.00
10.00
t (shakes)

The Code - 1

```
(* Define diffusion parameters. *)
Dn = 0.001;  (* self diffusion coefficient in m^2/shake *)
Cn = 1.0;   (* Creation rate in fraction/shake. *)
Ln = 0.13;  (* size of the region in meters. *)

(* parameters for the algorithm. *)
tmax = 10.0;  (* maximum time in shakes. *)
Nxsteps =   40;  (* steps in x. *)
Ntsteps = 3000;   (* steps in time. *)
dx = Ln/Nxsteps;  (* stepsize in x (m). *)
dt = tmax/Ntsteps;   (* stepsize in time (shakes). *)

(* set up the distribution of neutrons at t=0 so there is always
a spike of the same size in the middle. *)
n0 =  Table[{x, 0, 0}, {x, 0, Ln, dx}];
n0[[IntegerPart[Nxsteps/2], 3]] = 1/dx ;

(* some test parameters. *)
tsigma = dx^2/(2*Dn);
```

The Code - 2

```
(* monitor the choice of parameters. *)
Print["tsigma=", tsigma, " shakes, dt=", dt, " shakes, L=", Ln, " m"]

(* initialize the main array. *)
neutron = Table[0.0, {i, 1, Nxsteps}, {n, 1, Ntsteps}];
(* put in the initial conditions for t=0. *)
Do[neutron[[i, 1]] = n0[[i, 3]], {i, 1, Nxsteps}];
(* The boundary condition at x=0. *)
Do[neutron[[1, n]] = 0.0, {n, 2, Ntsteps}];
(* The boundary condition at x=L. *)
Do[neutron[[Nxsteps, n]] = 0.0, {n, 2, Ntsteps}];

(* constants for the recursion relation. *)
A0 = 1 - (2*dt*Dn)/dx^2 + dt*Cn;
B0 = (dt*Dn)/dx^2;
```

The Code - 3

```
(* main loop. outer loop over time and inner loop over position. *)
Do[
 Do[neutron[[i, n]] =
   A0*neutron[[i, n - 1]] + B0*neutron[[i + 1, n - 1]] +
    B0*neutron[[i - 1, n - 1]],
  {i, 2, Nxsteps - 1}](* end of inner loop *),
 {n, 2, Ntsteps}] (* end of outer loop *)

(* plotting the results in the middle of the x range. *)
xcounter = IntegerPart[Nxsteps/2];
xvalue = dx*xcounter;
t1 = Table[{dt*(n - 1), neutron[[xcounter, n]]}, {n, 1, Ntsteps}];
t1a = Table[t1[[n, 2]], {n, 2, Ntsteps}];
p1 = ListLogLogPlot[t1,
  PlotRange -> {{dt, tmax}, {Automatic, Automatic}}, Frame -> True,
  FrameLabel -> {"t (shakes)", "n (inverse meters)",
        StringForm["Neutron Diffusion, Ln=‘‘ m", Ln], ""}, Joined -> True
  BaseStyle -> Large, PlotStyle -> Thickness[0.005],
  LabelStyle -> Directive[Larger], ImageSize -> 7*72]
```