Having gotten Vaishakhi’s code working, we’re interpreting some of the results from the .arr graph.

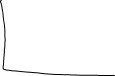
3 graphs:



* fixed range



* fixed depth across a range (like looking at attenuation)



* + continuous?



Initially doing 30 beams, running 300 beams –

How do they decide where the beams go?

John Dai’s graph models a beam stuck out in free space – not free space, but an empty ocean

Directivity information: alpha1, 2; beta 1, 2 a 2d system aimed at a 30 degree angle

Two modes – two times hitting the sensor at a particular range and time – from surface echo?

Each dot is a ray hitting the director,

Not reflected

Only happens after a while in time

Understanding the setup for the simulation and then try to create something

Simpler test scenario’

Deflection layer in the top of the ocean - each depth has a certain speed

Prof wants a simpler model – with no ocean – making the ocean super deep

Simpler model:

* No doubling (so that must have been from the echoes of the surface and ocean floor)
* Dying out faster
* Dots along a vertical line mark where the receiver got on the Gaussian beam as well as attenuation
* Changing across the range is different receivers, and what they get over time

Add one object to reflect off of

Changing the beam to point either on the top or on the bottom

To look at the impulse responses – one receiver examine the time evolution

How many rays does it take to create an interesting distribution

Slowly increasing the complexity of our models to compare with the hardware team

Sources and receivers stepping down – to understand movement

Alpha changes the direction of the collection of rays: 75 90 to shine down

Plotarr measures the arrivals for one specific receiver

Looking at time response we can see the flat surface

1. Notes
2. Adding reflection to the bottom
3. Creating the impulse response from the .arr file
4. Adding features to the bottom of the ocean