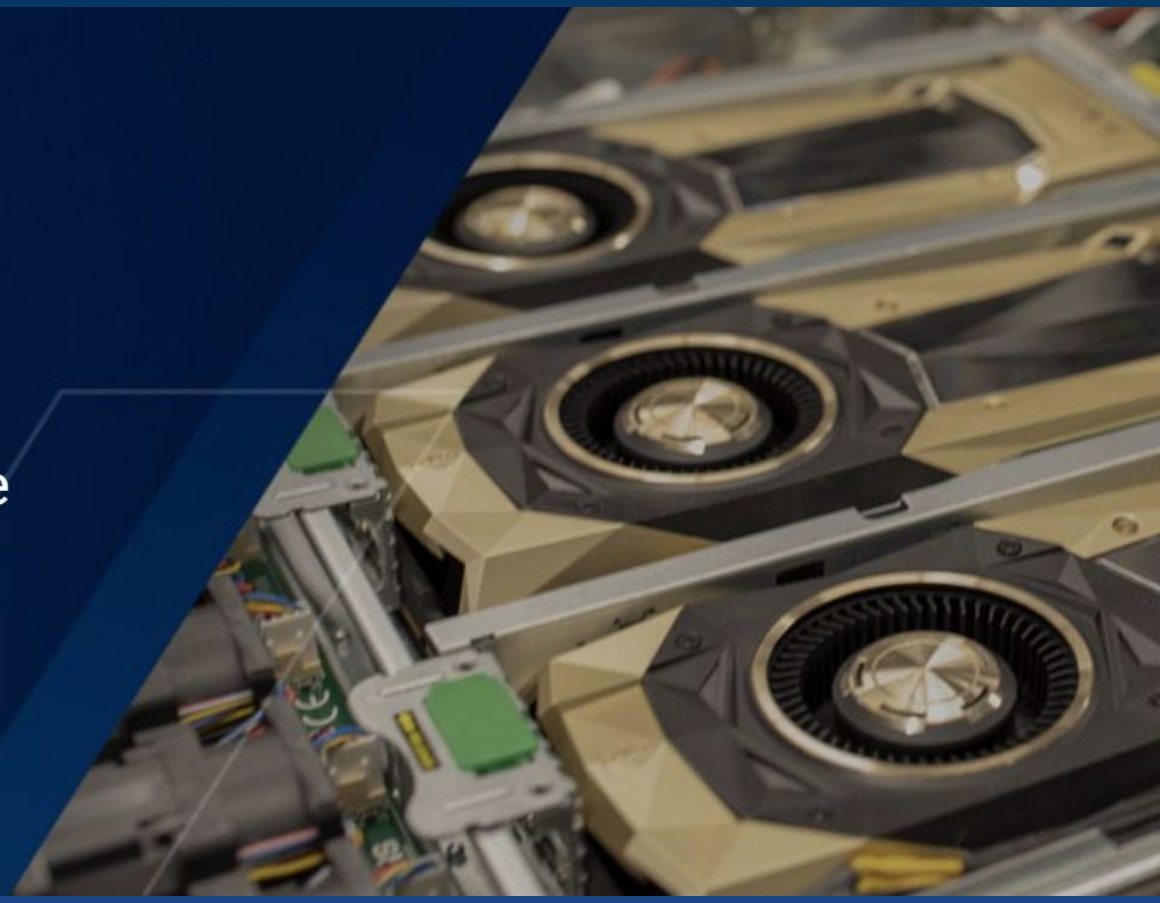
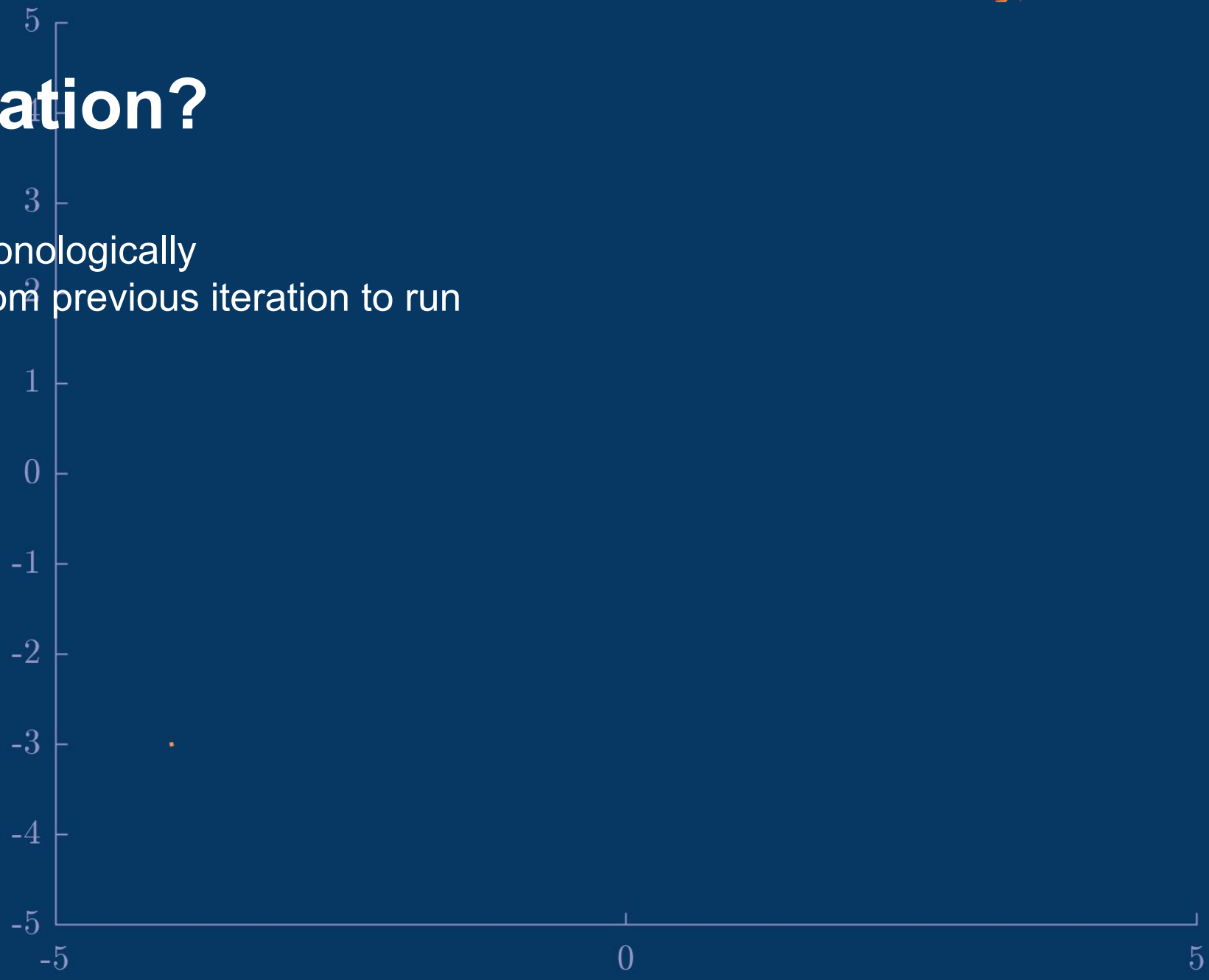
Parallel Computing Toolbox

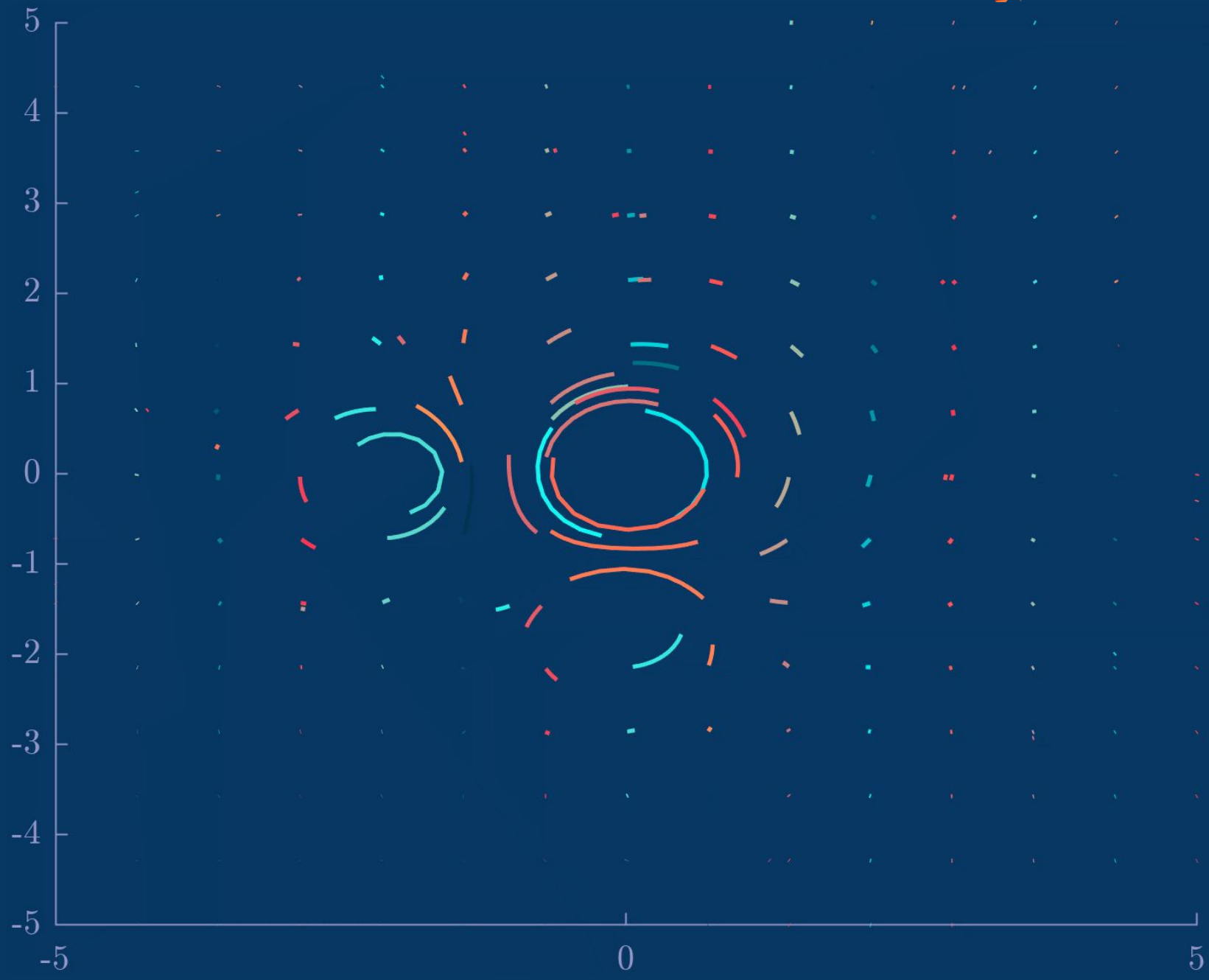
Perform parallel computations on multicore computers, GPUs, and clusters



What is parallelization?

- Most MATLAB scripts run chronologically
- for-loops, need information from previous iteration to run




5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
0
5

Problems suitable for parallelization:

- Problems where one process is not dependant on the other
- No (*) need for communication between processes

Why parallelize?

Well, depends on your problem!

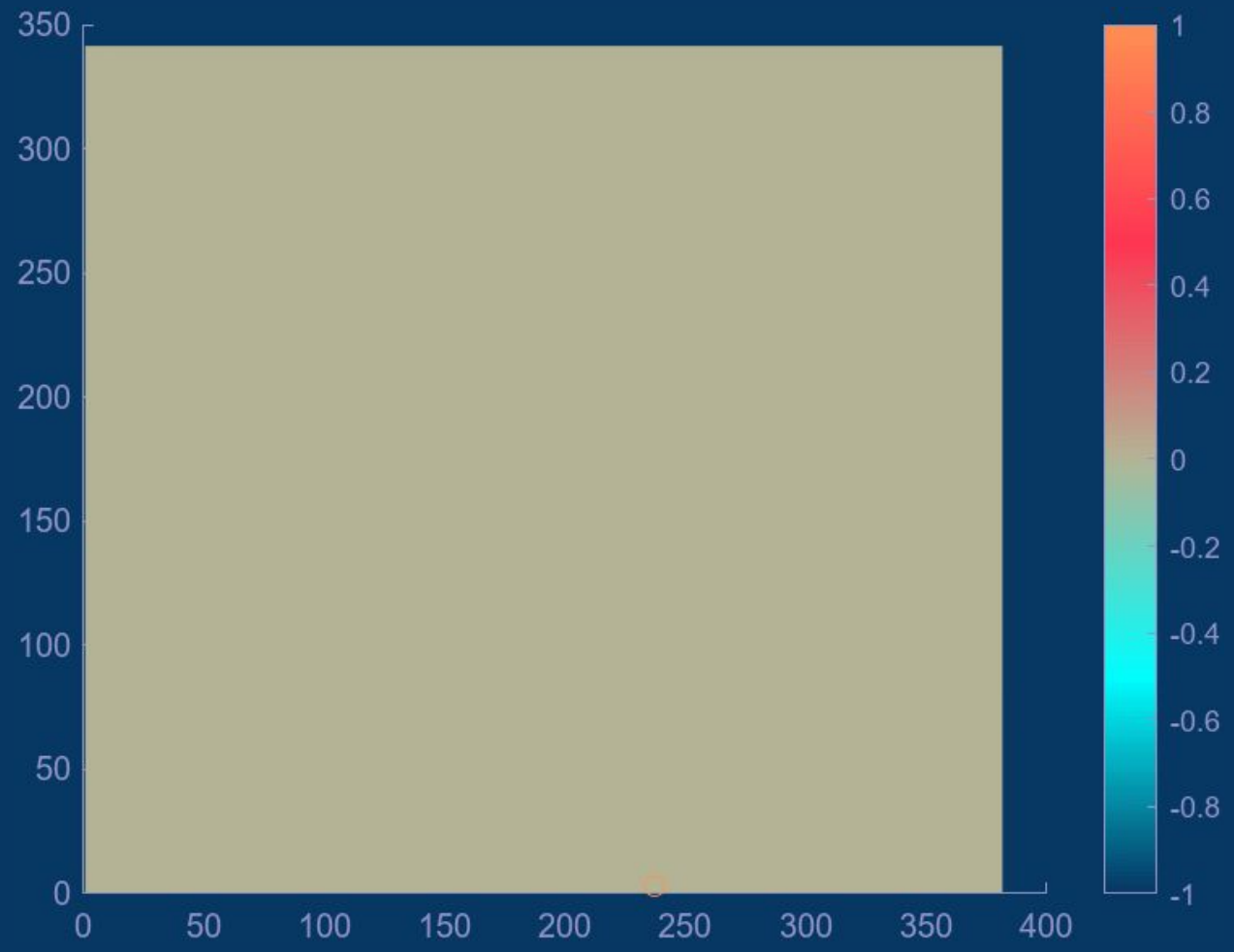
- SPEED AND POWER!!!

Different ways of running MATLAB in parallel:

- parfor
- arrayfun, gpuArray
- parfeval
- tall arrays
- batch

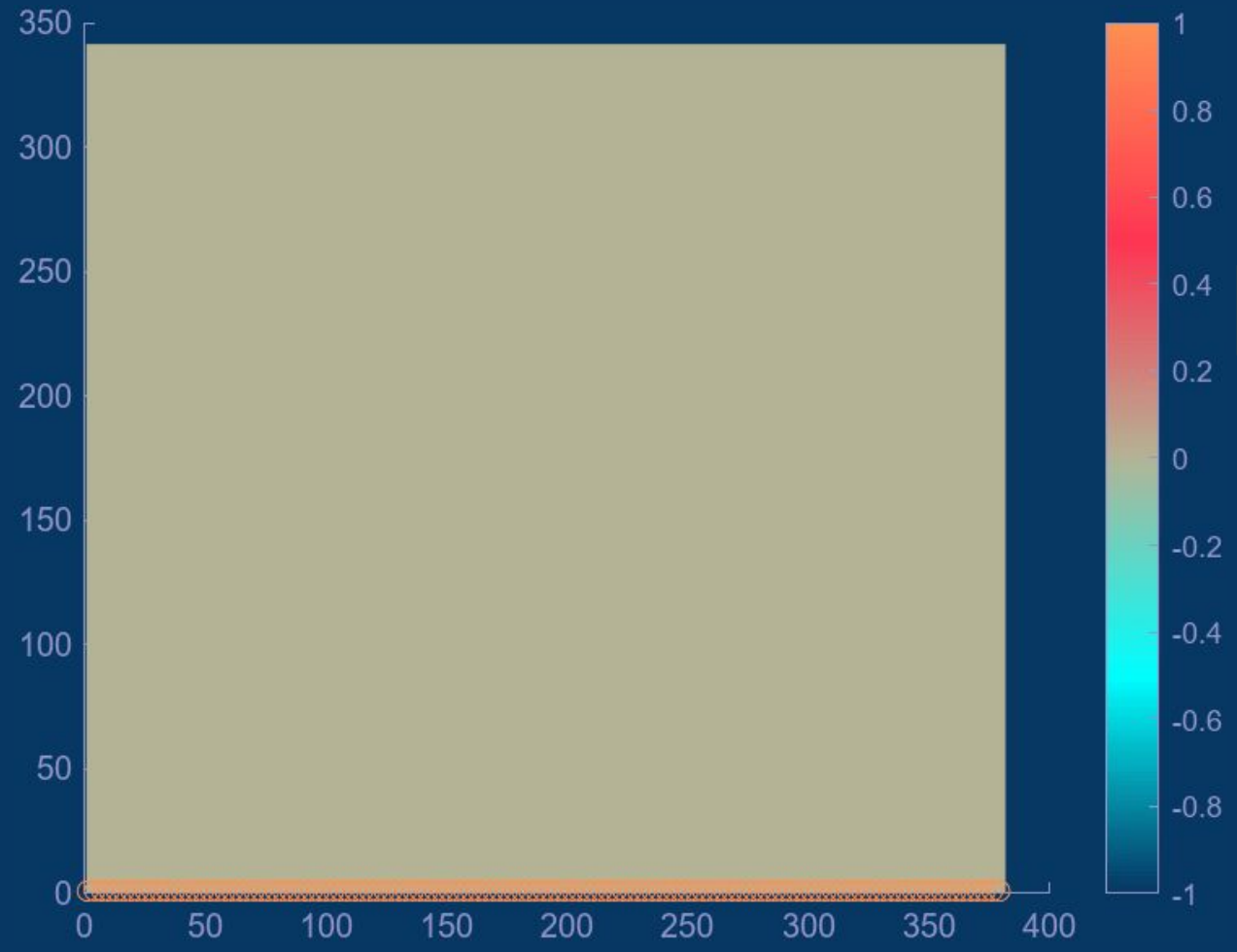
```
parfor
arrayfun()
gpuArray()
parfeval()
batch()
```

Normal for-loop:




```
X = -2.3:0.01:1.5;
Y = -1.7:0.01:1.7;
iterations = zeros(width(Y), width(X));



for y_index = 1:width(Y)
for x_index = 1:width(X)

    c = X(x_index) + Y(y_index)*1i;
    z = c;
    iteration = 0;

    while true
    z = z^2 + c;

    if iteration > 1000 | norm(z) > 2; break; end

    iteration = iteration +1;
    end

    if iteration < 100; iterations(y_index, x_index) = iteration; end


end
end
```

parfor-loop:




```
X = -2.3:0.01:1.5;
Y = -1.7:0.01:1.7;
iterations = zeros(width(Y), width(X));



for y_index = 1:width(Y)
parfor x_index = 1:width(X)

    c = X(x_index) + Y(y_index)*1i;
    z = c;
    iteration = 0;

    while true
    z = z^2 + c;

    if iteration > 1000 | norm(z) > 2; break; end

    iteration = iteration +1;
    end

    if iteration < 100; iterations(y_index, x_index) = iteration; end
end
end
```

parfor-loop:

```
A = 1:2:100;
B = rand(100);
C = zeros(100);

parfor i=A

C(i,i) = B(A(i));


end
```

parfor-loop:

```
A = 1:2:100;
B = rand(100);
C = zeros(100);

parfor i=A
```

The PARFOR loop cannot run due to the way variable 'C' is used. Details ▼

```
end
```

parfor-loop:

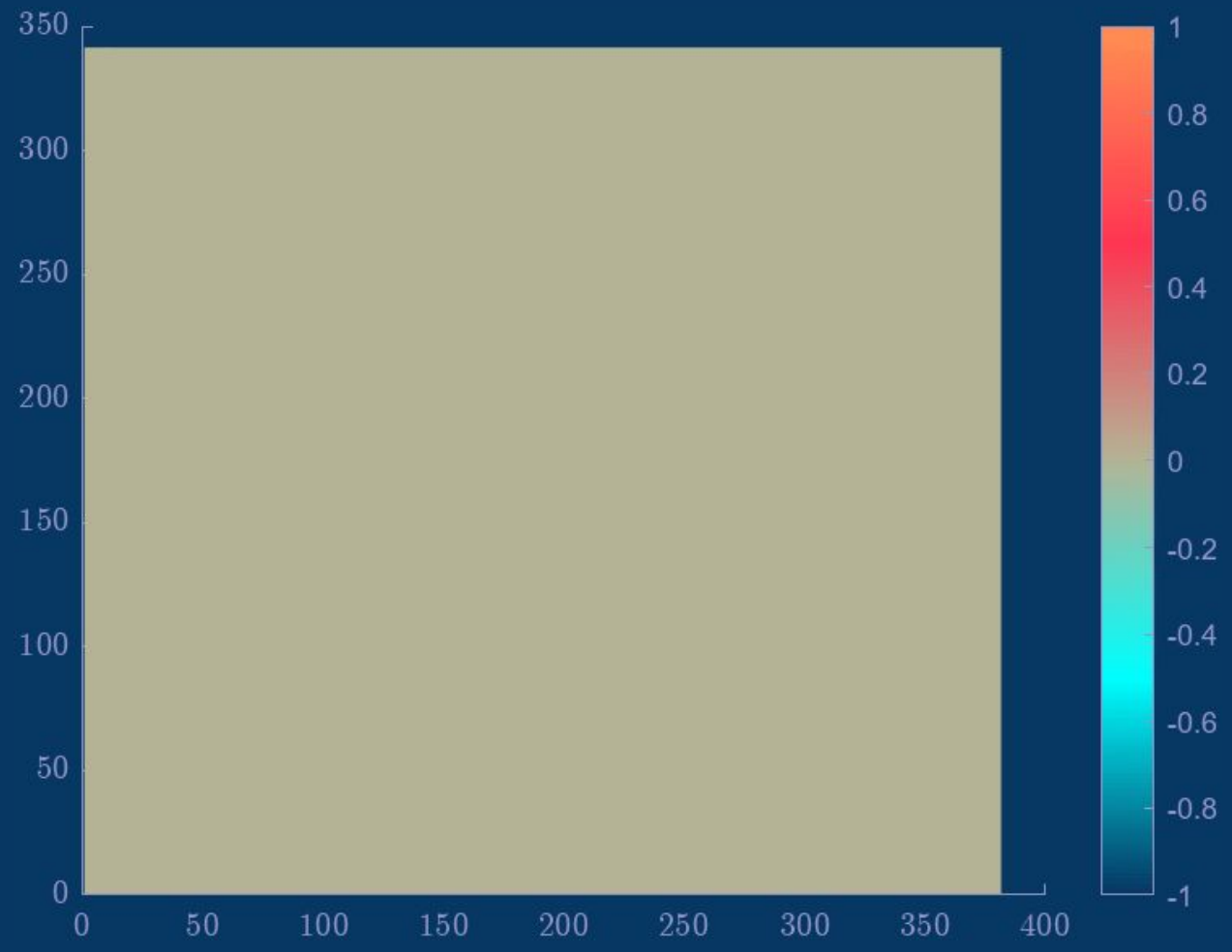

```
X = -2.3:0.01:1.5;
Y = -1.7:0.01:1.7;
iterations = zeros(width(Y), width(X));
C = X + Y'*1i;
```

```
iterations = reshape(iterations,1,[]);
C          = reshape(C,1,[]);
```

```
parfor index = 1:width(C)

    c = C(index);
    z = c;
    iteration = 0;

    while true
    z = z^2 + c;

    if iteration > 1000 | norm(z) > 2; break; end

    iteration = iteration +1;
    end

    if iteration < 100; iterations(index) = iteration; end
end
```

```
iterations = reshape(iterations, width(Y), []);
```

Vectorizing and arrayfun



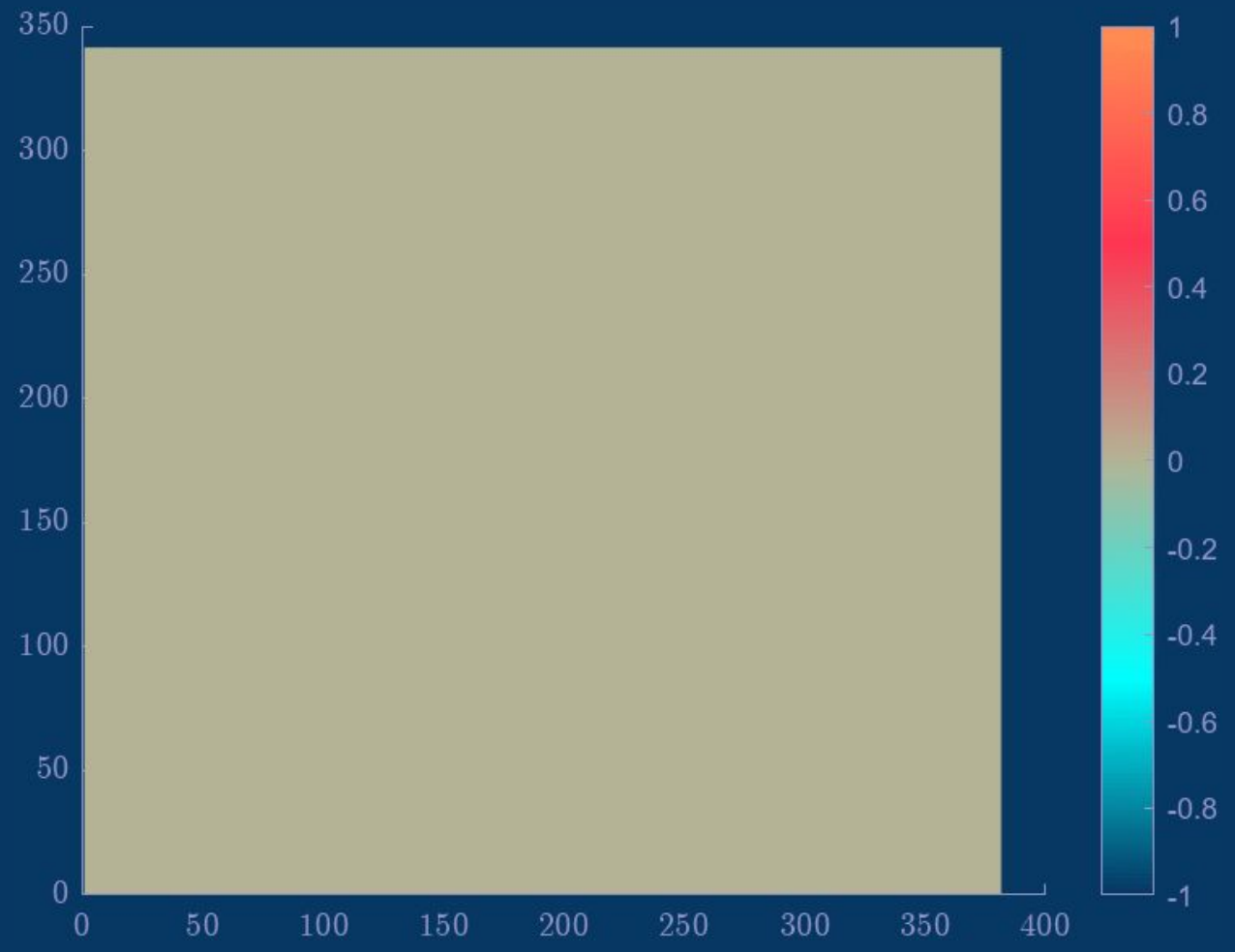

```
X = -2.3:0.01:1.5;
Y = -1.7:0.01:1.7;

C = X + Y'*1i;

iterations = arrayfun(@evaluate_iterations, C);
```

```
function iteration = evaluate_iterations(c)

    z = c;
    iteration = 0;

    while 100 > iteration && 2 > norm(z)

    z = z^2 + c;
    iteration = iteration +1;
    end

    if iteration == 100; iteration = 0; end
end
```

Vectorizing and arrayfun

```
A = rand(100);
B = rand(100);

C   =  A +B;
D   =  A.*B;
E   =  arrayfun(  @(a,b) a + b,        A,B)
```

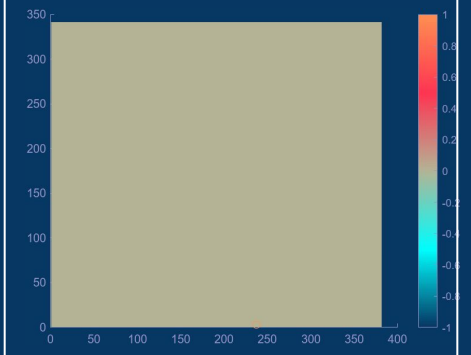
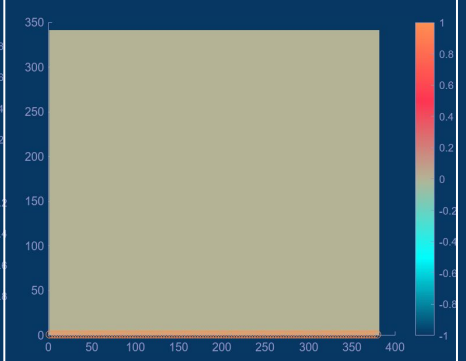
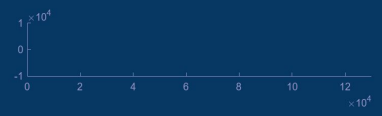
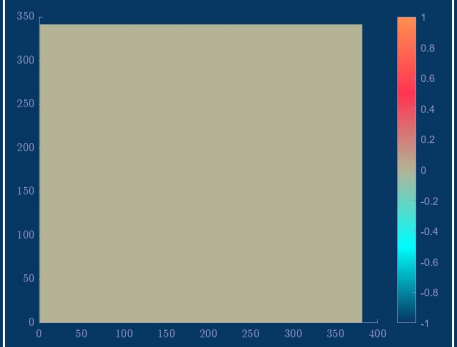
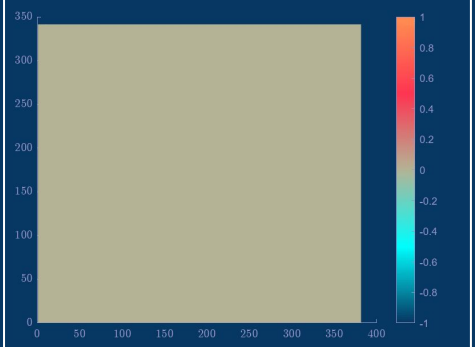
gpuArray:



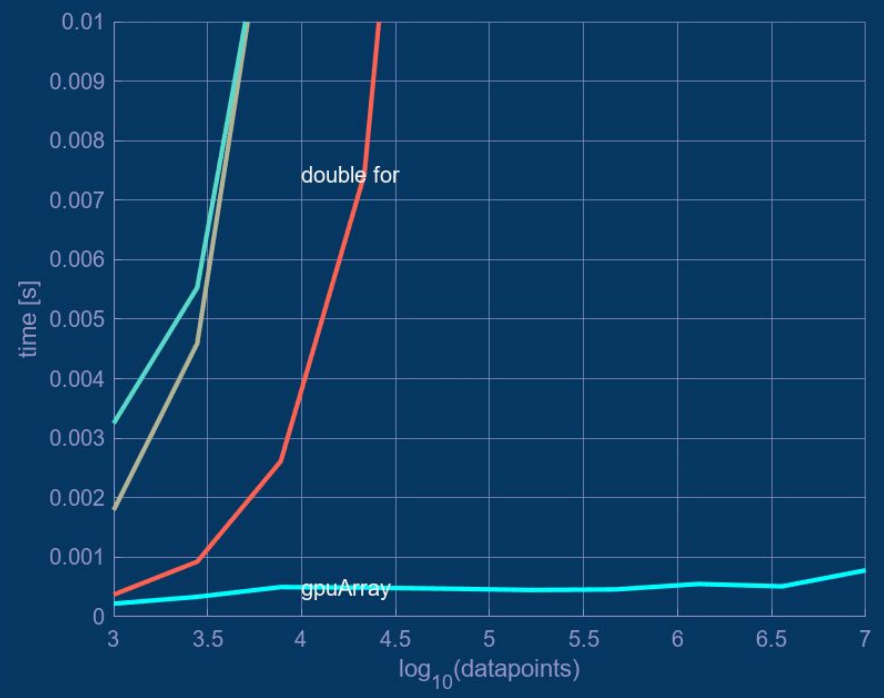

```
X = -2.3:0.01:1.5;
Y = -1.7:0.01:1.7;

C = X + Y'*1i;
C = gpuArray(C);

iterations = arrayfun(@evaluate_iterations, C);
```

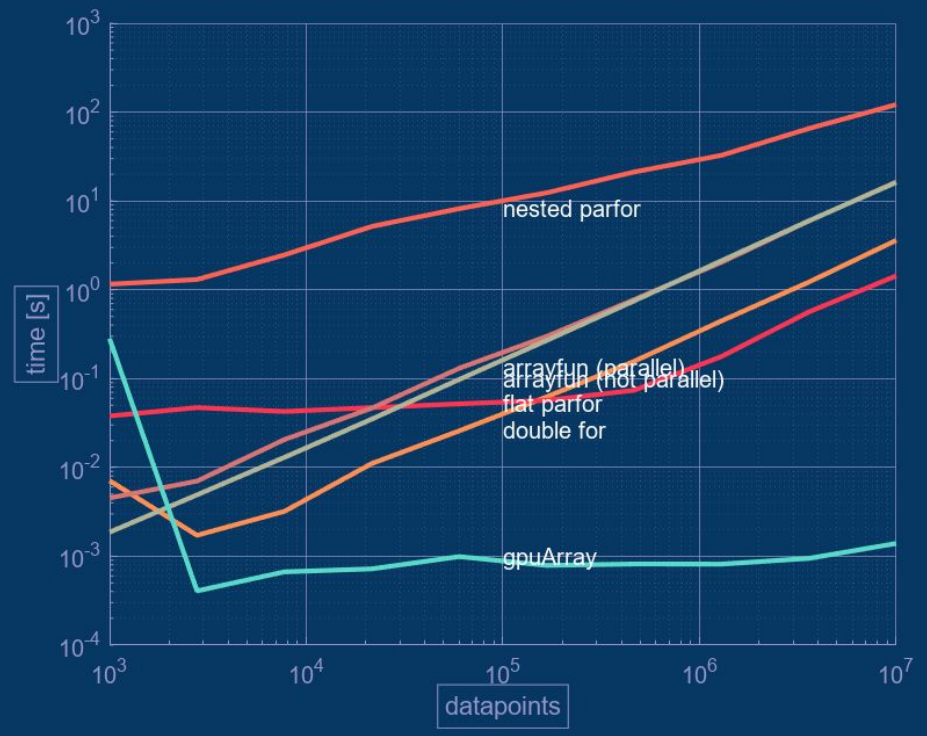
```
function iteration = evaluate_iterations(c)

    z = c;
    iteration = 0;

    while 100 > iteration && 2 > norm(z)

    z = z^2 + c;
    iteration = iteration +1;
    end

    if iteration == 100; iteration = 0; end
end
```

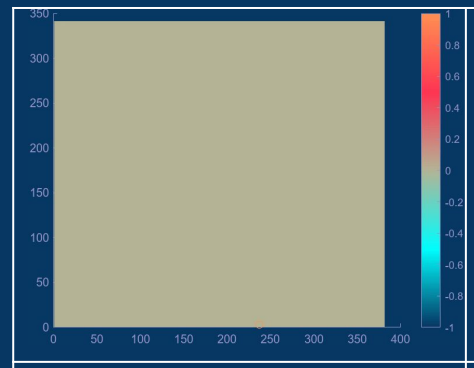
Benchmark 10^5 datapoints:

					
	Double for-loop	**Nested parfor**	**Flat parfor**	**Arrayfun**	**Arrayfun, gpuArray**
s	0.026	8.192	0.052	0.097 (not parallel) 0.1306 (parallel)	0.00099
ms	26	8 192	52	97 130	0.99

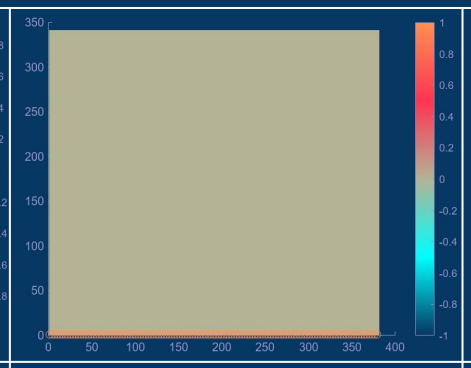
Computer specs:
Processor: 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz, 2496 Mhz, 8 Core(s), 16 Logical Processor(s)
GPU: NVIDIA Quadro P1000
Ram: 32 GB



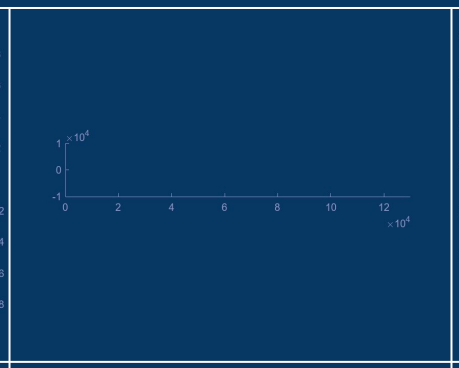




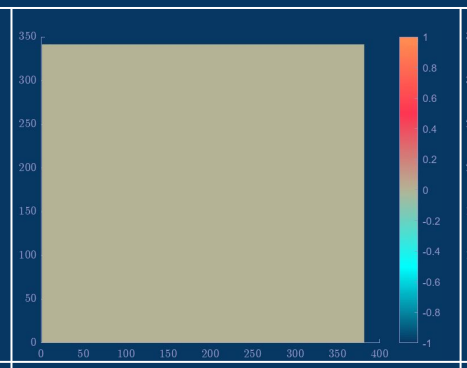
Double for-loop



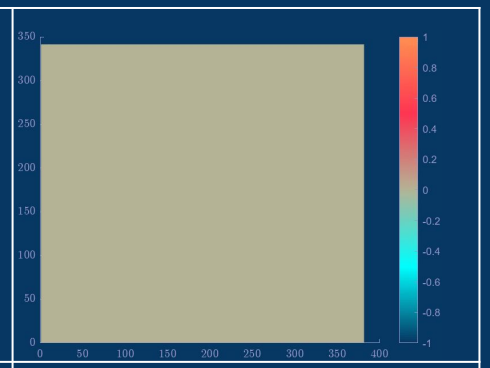
Nested parfor



Flat parfor



Arrayfun



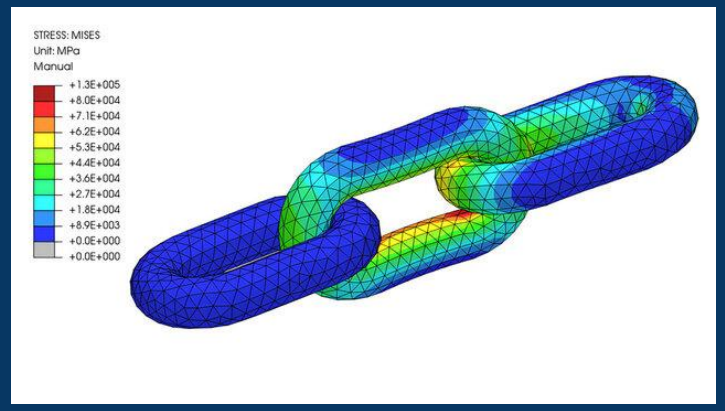
Arrayfun, gpuArray

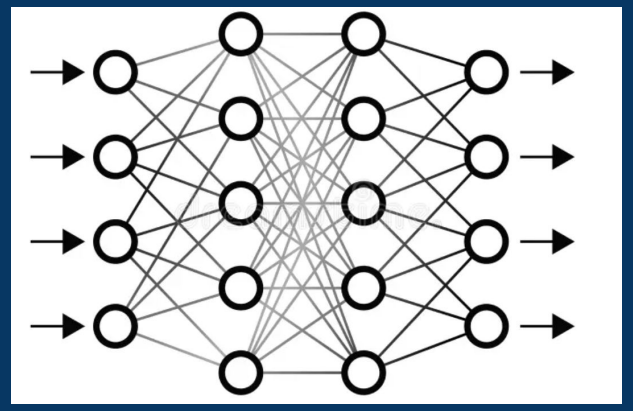
Applications?

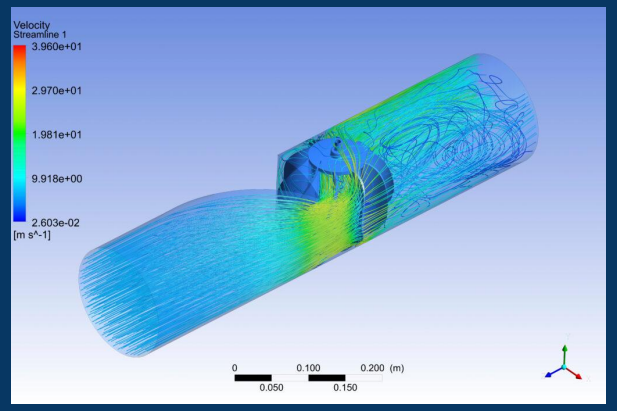
Applications?

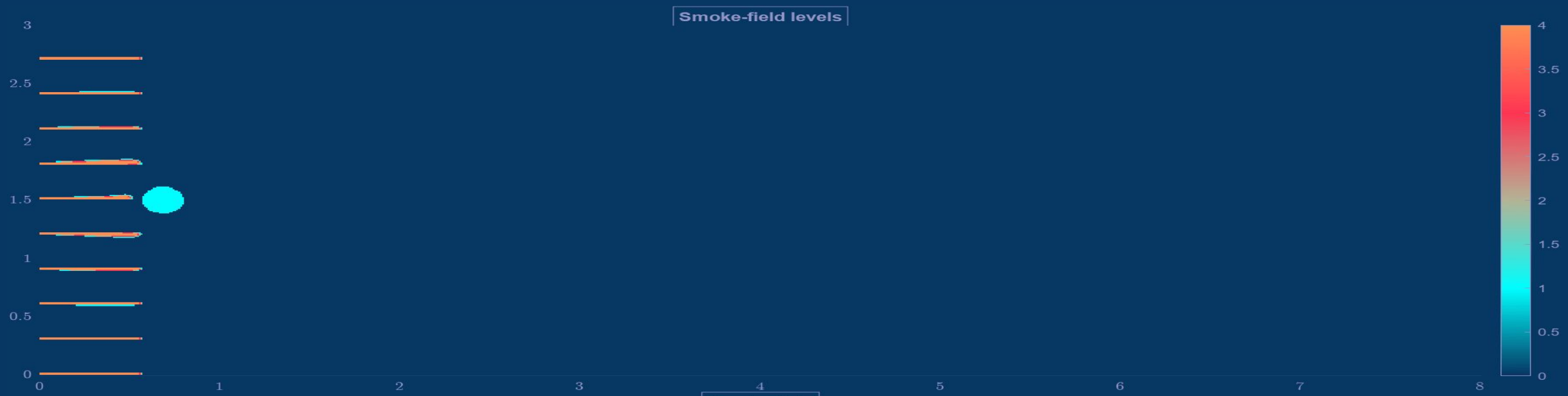
…Well there are many!

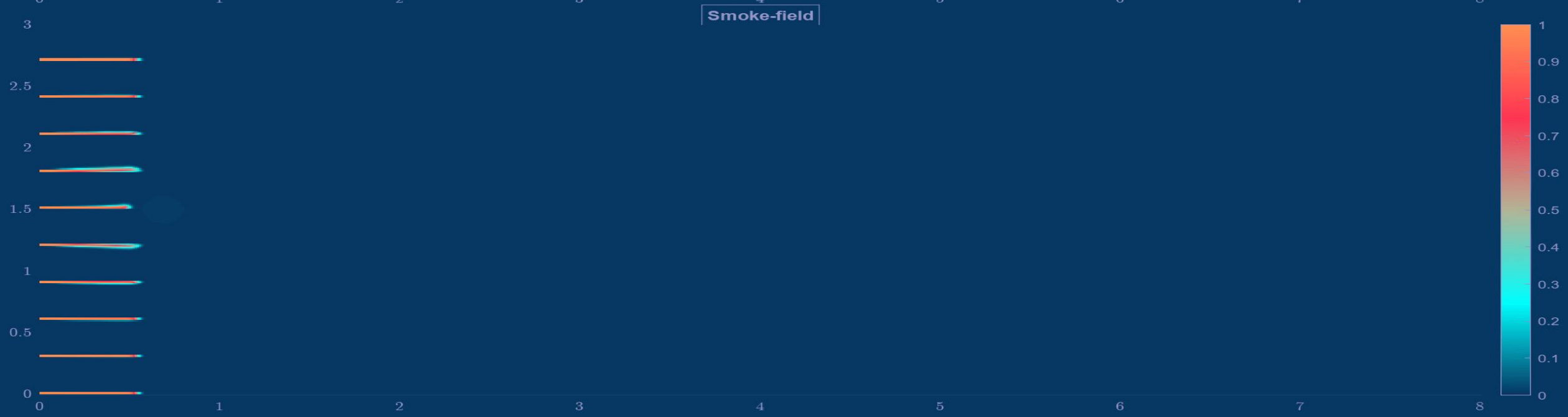
- AI, neural networks & machine learning
- Graphics
- Solving matrix-problems
- Fluid mechanics, CFD





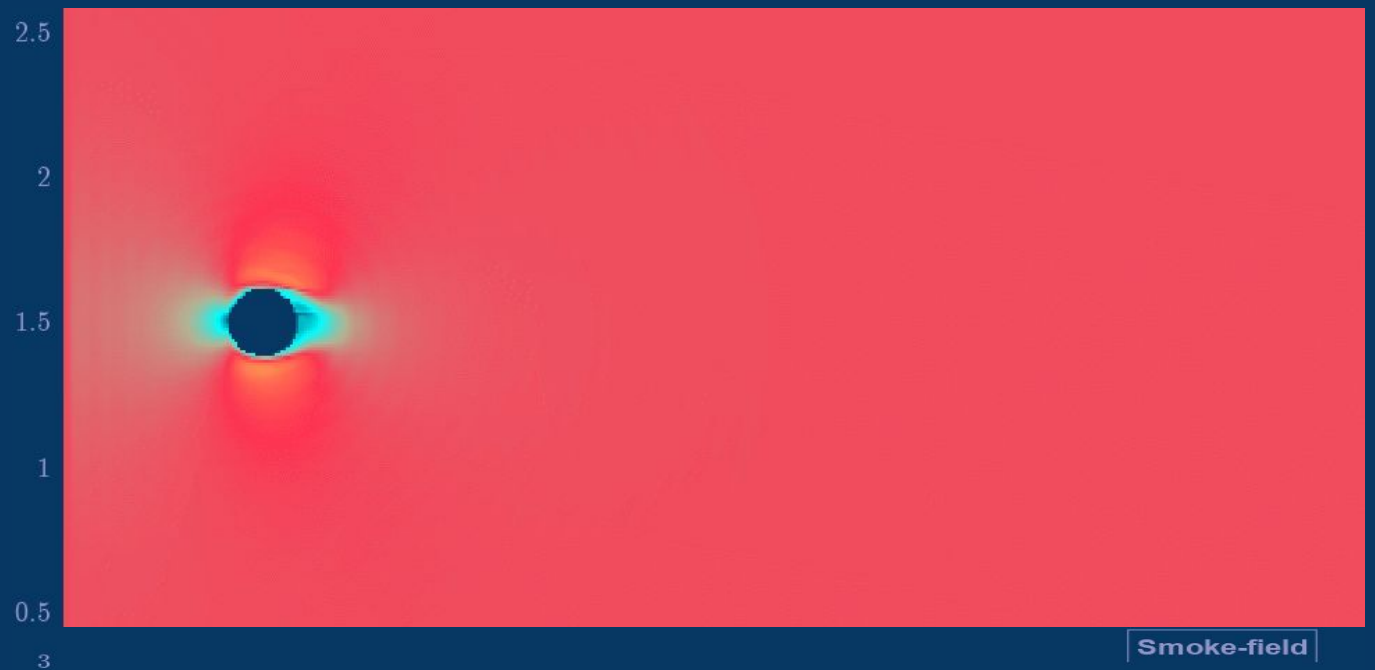


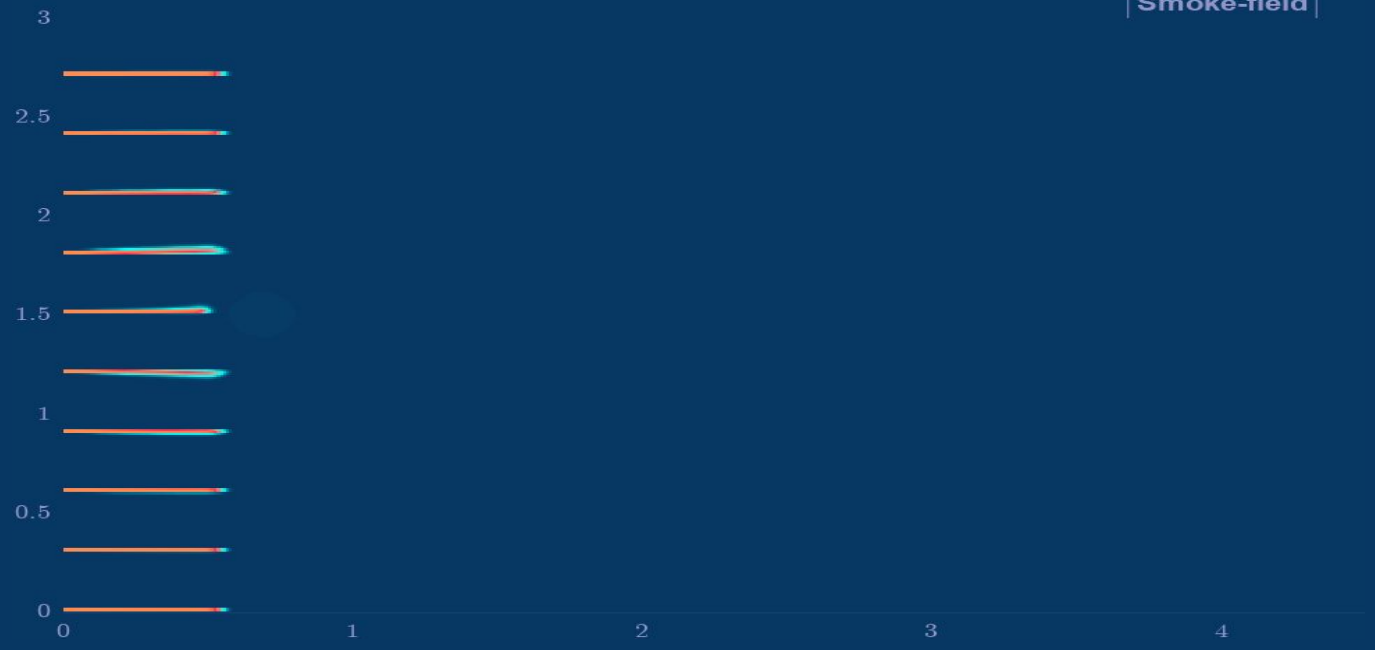

Smoke-field levels
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
4
3.5
3
2.5
2
1.5
1
0.5
0


Smoke-field
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


2.5
2
1.5
1
0.5
1.2
1
0.8
0.6
0.4
Smoke-field


3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0





With gpuArray:

0.0778 s/iteration **= 77.84 ms/it**

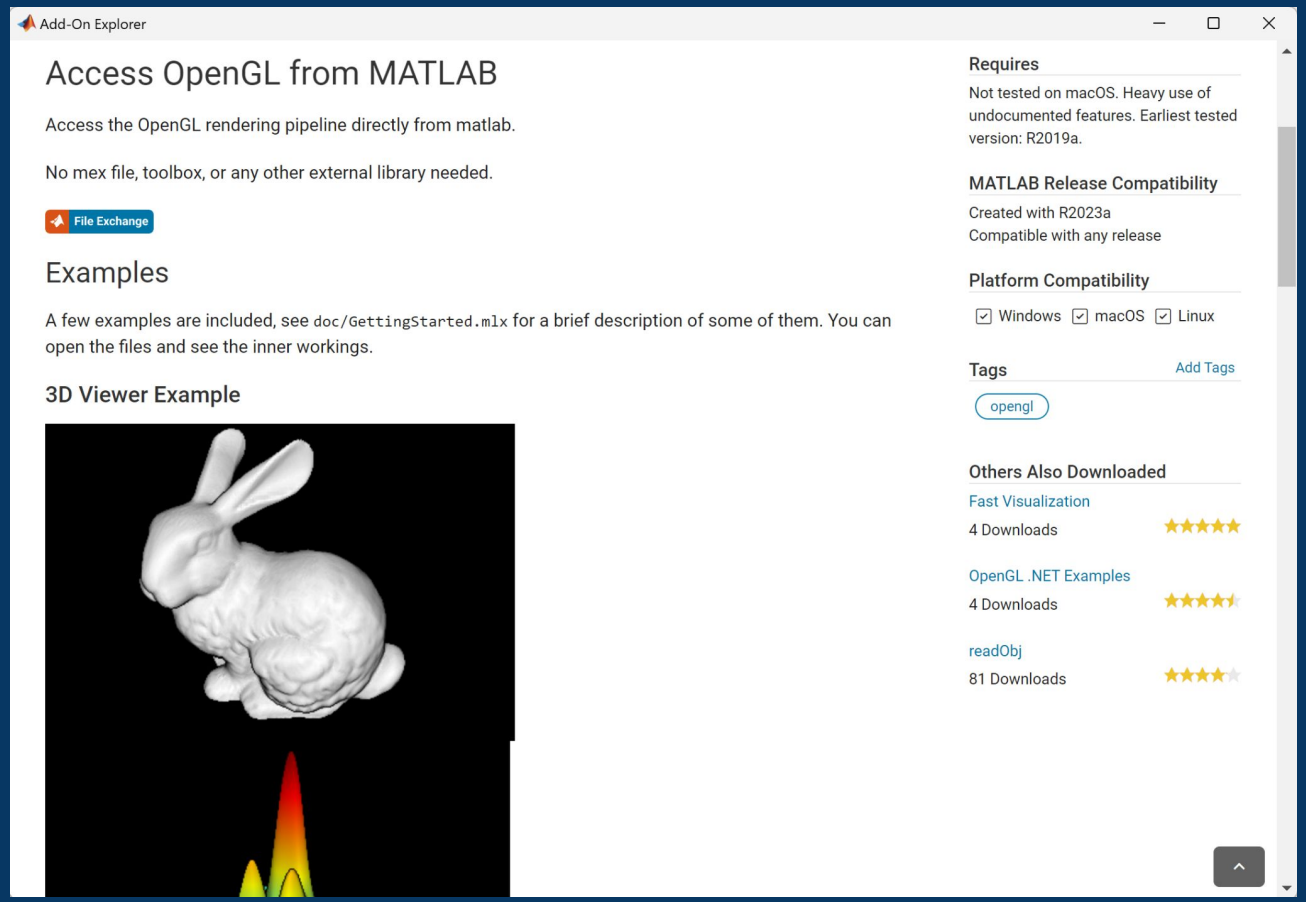
Without gpuArray:

1.7432 s/iteration **= 1743.2 ms/it**

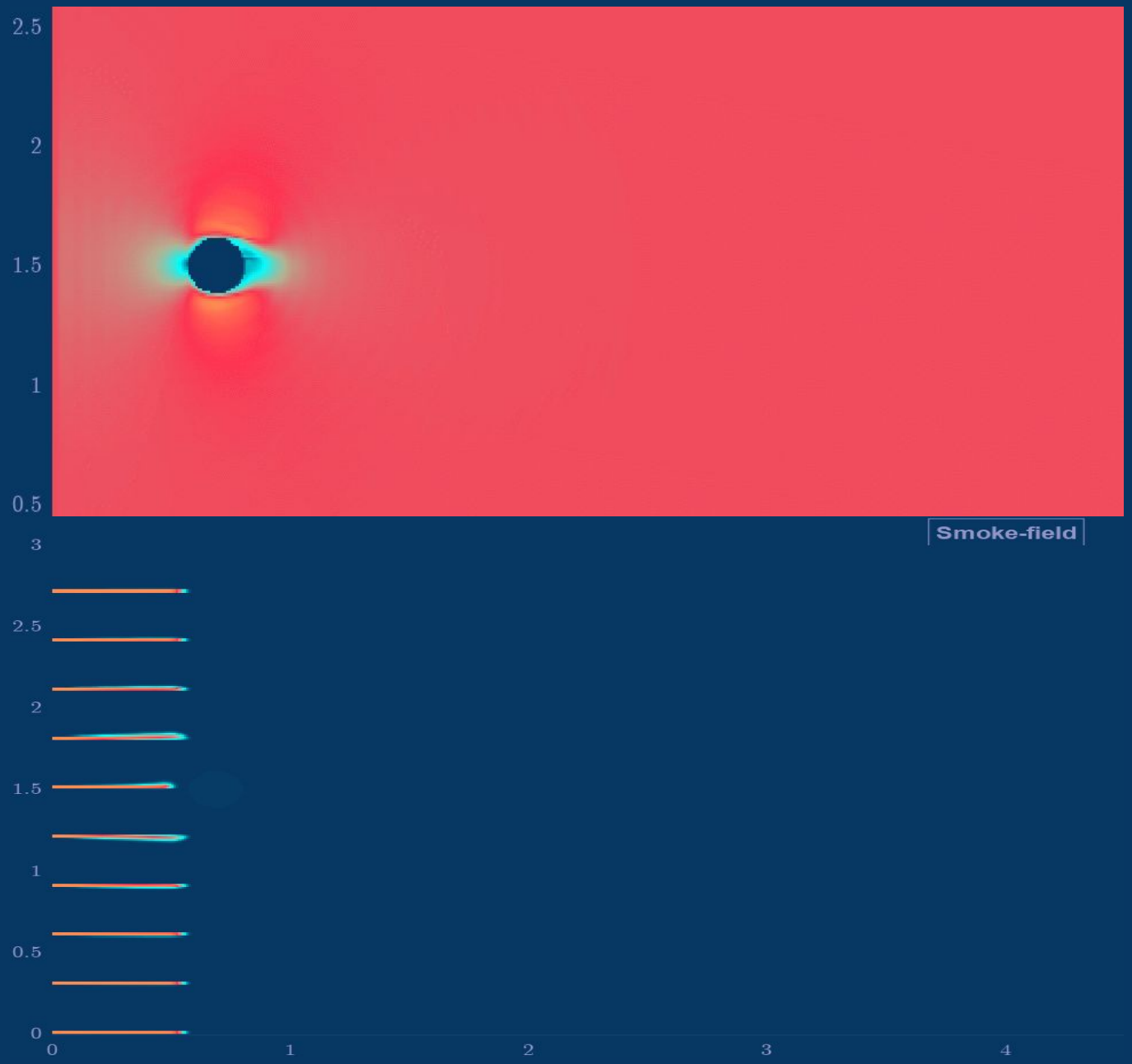
(we want the parallel computing toolbox to support Intel and AMD GPUs!!! >:( )

Access OpenGL from MATLAB

A standalone library developed by community-member **Francis Plamondon**



https://github.com/fr0nkk/matogl




```
for iteration = 1:max_iterations

projection
staggered2centered
advection
viscosity
centered2staggered
stabilize

end
```

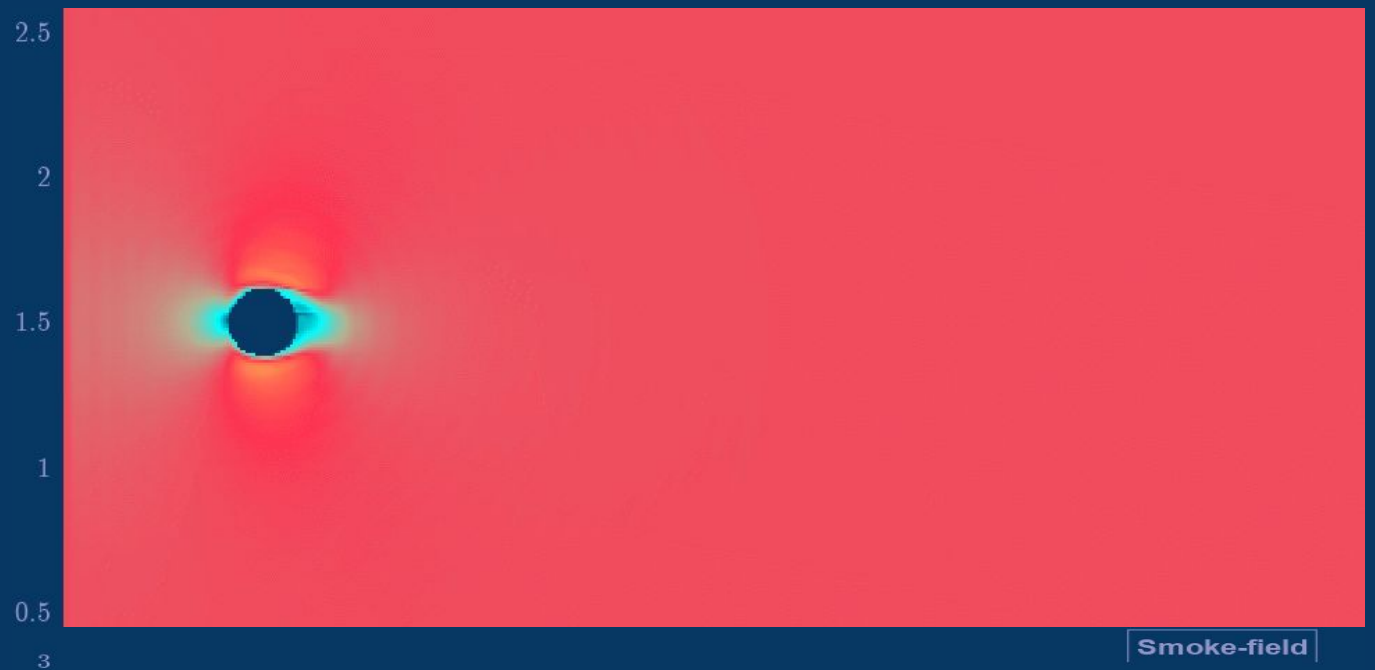
projection

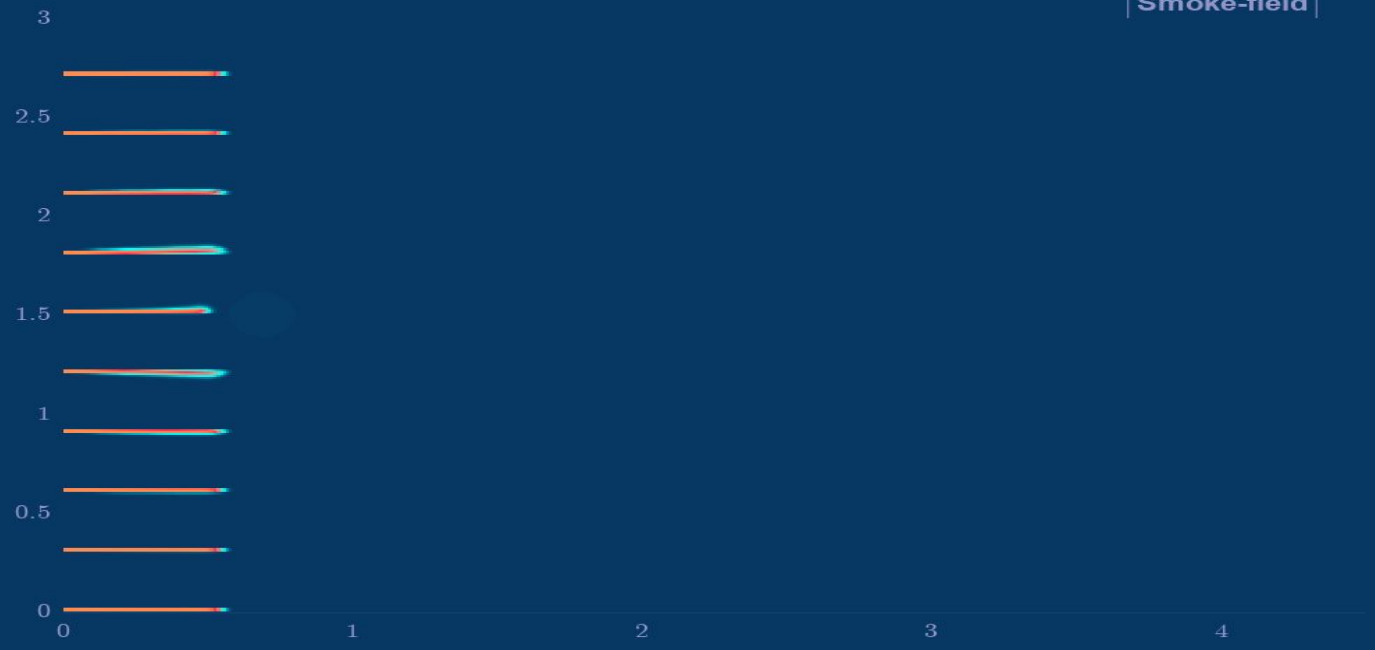
↓

advection

↓

viscosity





The project is up on **Github** and the **MATLAB file-exchange** for anyone interested!

(Just don't use it for anything important bcs it kinda bad 🫠🫠🫠)

https://github.com/spiggen/MATLAB-fluid-sim

Parallel Computing with MATLAB: Hands-on workshop

24th of April 2024, 13:00

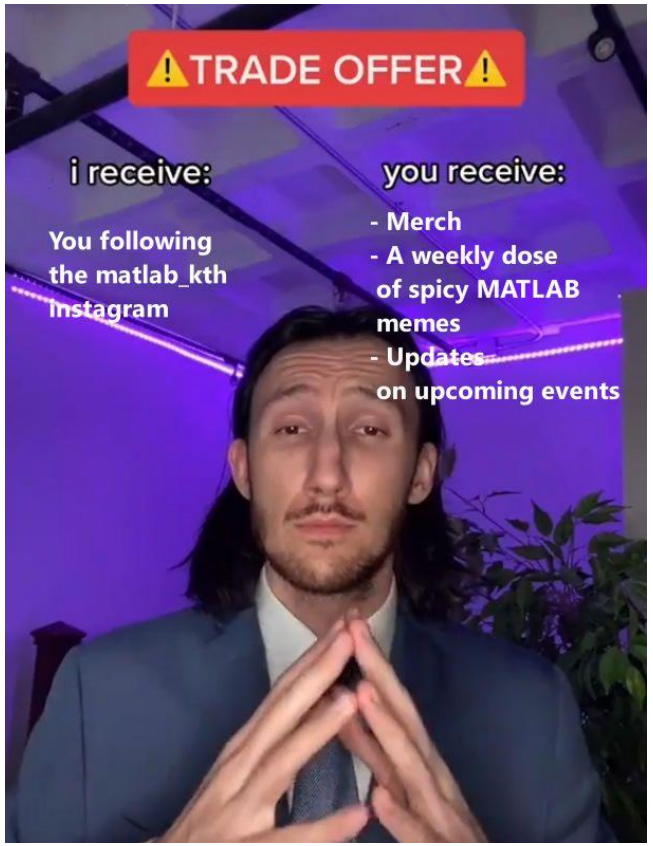
- Multithreading vs multiprocessing
- When to use parfor vs parfeval constructs
- Creating data queues for data transfer
- Leveraging NVIDIA GPUs
- Parallelizing Simulink models
- Working with large data

tinyurl.com/ParallelComputing24

Merch giveaway requirements:

- Follow matlab_kth on instagram or join the facebook-group!

matlab_kth
facebook
group



SCAN ME

matlab_kth
instagram