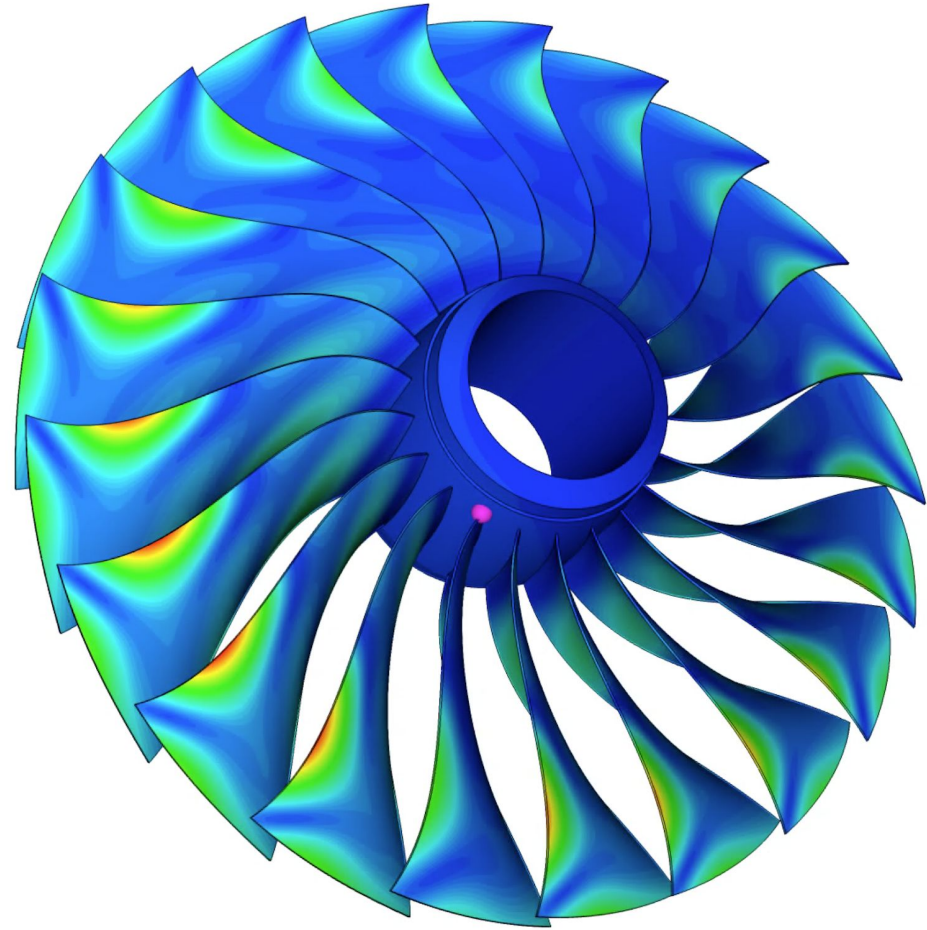


GPU accelerated computing for Finite Element Method



Introduction to GPU

Heterogenous program

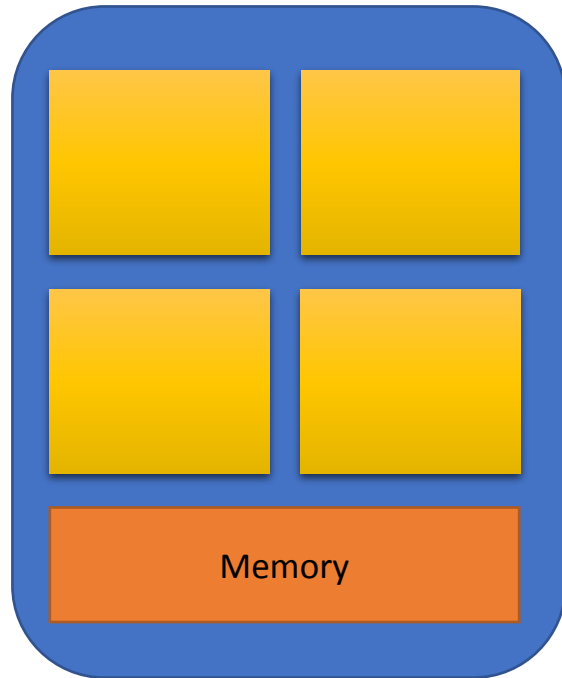
Objectives

- To learn the basics of CUDA code
 - Host and Device
 - Memory allocation
 - Data transfer

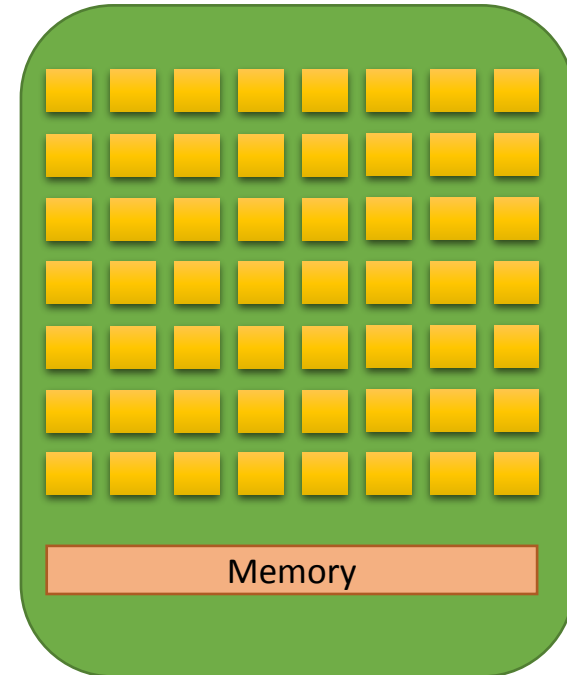
Introduction to GPU

Heterogenous program

CPU - Host



GPU – Device



Introduction to GPU

Heterogenous program

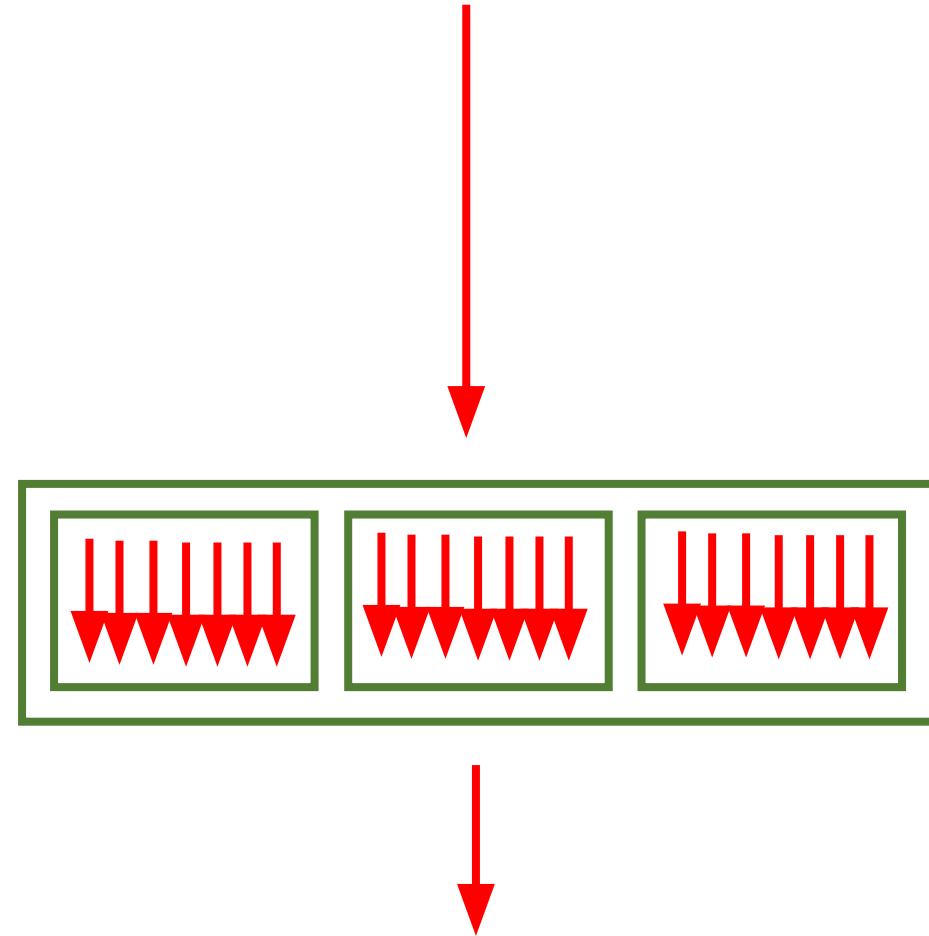
CUDA programming Model – Heterogenous Computing

```
16 int main(){
17     float *a, *b, *out;
18     float *d_a, *d_b, *d_out;
19
20     a = (float*)malloc(sizeof(float) * N);
21     b = (float*)malloc(sizeof(float) * N);
22     out = (float*)malloc(sizeof(float) * N);
23
24     // Initialize array
25     for(int i = 0; i < N; i++){
26         a[i] = 1.0f;
27         b[i] = 2.0f;
28     }
29
30     // Allocate device memory for a
31     cudaMalloc((void**)&d_a, sizeof(float)*N);
32     cudaMalloc((void**)&d_b, sizeof(float)*N);
33     cudaMalloc((void**)&d_out, sizeof(float)*N);
34
35     // Transfer data from host to device memory
36     cudaMemcpy(d_a, a, sizeof(float)*N, cudaMemcpyHostToDevice);
37     cudaMemcpy(d_b, b, sizeof(float)*N, cudaMemcpyHostToDevice);
38
39     // Main function
40     int block_size = 256;
41     int grid_size = (N+block_size)/block_size;
42     vector_add<<<grid_size, block_size>>>>(d_out, d_a, d_b, N);
43
44     cudaMemcpy(out, d_out, sizeof(float)*N, cudaMemcpyDeviceToHost);
45
46     // Deallocate device memory
47     cudaFree(d_a);
48     cudaFree(d_b);
49     cudaFree(d_out);
50
51     // Deallocate host memory
52     free(a);
53     free(b);
54     free(out);
55
56 }
```

Serial Code

Parallel Code

Serial Code



Serial Code
in host

Parallel Code
in device

Serial Code
in host

Introduction to GPU

Heterogenous program

CUDA programming Model – Heterogenous Computing

Host – CPU, Device - GPU

```
16 int main(){
17     float *a, *b, *out;
18     float *d_a, *d_b, *d_out;
19
20     a = (float*)malloc(sizeof(float) * N);
21     b = (float*)malloc(sizeof(float) * N);
22     out = (float*)malloc(sizeof(float) * N);
23
24     // Initialize array
25     for(int i = 0; i < N; i++){
26         a[i] = 1.0f;
27         b[i] = 2.0f;
28     }
29
30     // Allocate device memory for a
31     cudaMalloc((void**)&d_a, sizeof(float)*N);
32     cudaMalloc((void**)&d_b, sizeof(float)*N);
33     cudaMalloc((void**)&d_out, sizeof(float)*N);
34
35     // Transfer data from host to device memory
36     cudaMemcpy(d_a, a, sizeof(float)*N, cudaMemcpyHostToDevice);
37     cudaMemcpy(d_b, b, sizeof(float)*N, cudaMemcpyHostToDevice);
38
39     // Main function
40     int block_size = 256;
41     int grid_size = (N+block_size)/block_size;
42     vector_add<<<grid_size, block_size>>>>(d_out, d_a, d_b, N);
43
44     cudaMemcpy(out, d_out, sizeof(float)*N, cudaMemcpyDeviceToHost);
45
46     // Deallocate device memory
47     cudaFree(d_a);
48     cudaFree(d_b);
49     cudaFree(d_out);
50
51     // Deallocate host memory
52     free(a);
53     free(b);
54     free(out);
55
56 }
```

```
16 int main(){
17     float *a, *b, *out;
18     float *d_a, *d_b, *d_out;
19
20     a = (float*)malloc(sizeof(float) * N);
21     b = (float*)malloc(sizeof(float) * N);
22     out = (float*)malloc(sizeof(float) * N);
23
24     // Initialize array
25     for(int i = 0; i < N; i++){
26         a[i] = 1.0f;
27         b[i] = 2.0f;
28     }
29
30     // Allocate device memory for a
31     cudaMalloc((void**)&d_a, sizeof(float)*N);
32     cudaMalloc((void**)&d_b, sizeof(float)*N);
33     cudaMalloc((void**)&d_out, sizeof(float)*N);
34
35     // Transfer data from host to device memory
36     cudaMemcpy(d_a, a, sizeof(float)*N, cudaMemcpyHostToDevice);
37     cudaMemcpy(d_b, b, sizeof(float)*N, cudaMemcpyHostToDevice);
38
39     // Main function
40     int block_size = 256;
41     int grid_size = (N+block_size)/block_size;
42     vector_add<<<grid_size, block_size>>>>(d_out, d_a, d_b, N);
43
44     cudaMemcpy(out, d_out, sizeof(float)*N, cudaMemcpyDeviceToHost);
45
46     // Deallocate device memory
47     cudaFree(d_a);
48     cudaFree(d_b);
49     cudaFree(d_out);
50
51     // Deallocate host memory
52     free(a);
53     free(b);
54     free(out);
55
56 }
```

Memory Allocation
in Host

```
--
30 // Allocate device memory for a
31 cudaMalloc((void**)&d_a, sizeof(float)*N);
32 cudaMalloc((void**)&d_b, sizeof(float)*N);
33 cudaMalloc((void**)&d_out, sizeof(float)*N);
34
```

Memory Allocation
in Device

```
35 // Transfer data from host to device memory
36 cudaMemcpy(d_a, a, sizeof(float)*N, cudaMemcpyHostToDevice);
37 cudaMemcpy(d_b, b, sizeof(float)*N, cudaMemcpyHostToDevice);
--
```

Data transfer from
Host to Device

```
--
46 // Deallocate device memory
47 cudaFree(d_a);
48 cudaFree(d_b);
49 cudaFree(d_out);
50
51 // Deallocate host memory
52 free(a);
53 free(b);
54 free(out);
55
56
```

Deallocation of
Memory

```
cudaMemcpy(out, d_out, sizeof(float)*N, cudaMemcpyDeviceToHost);
```

Data transfer from
Device to Host

```
--
39 // Main function
40 int block_size = 256;
41 int grid_size = (N+block_size)/block_size;
42 vector_add<<<grid_size, block_size>>>>(d_out, d_a, d_b, N);
--
```

Computation in
Device