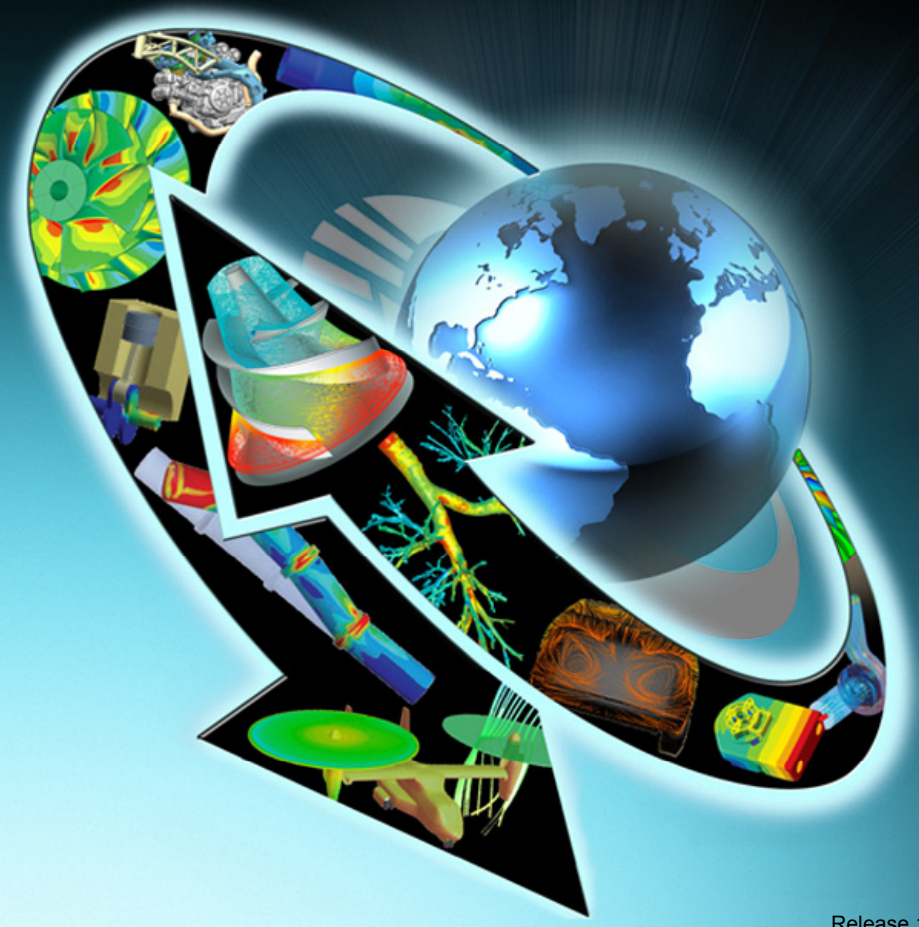
ANSYS®

Chapter 6

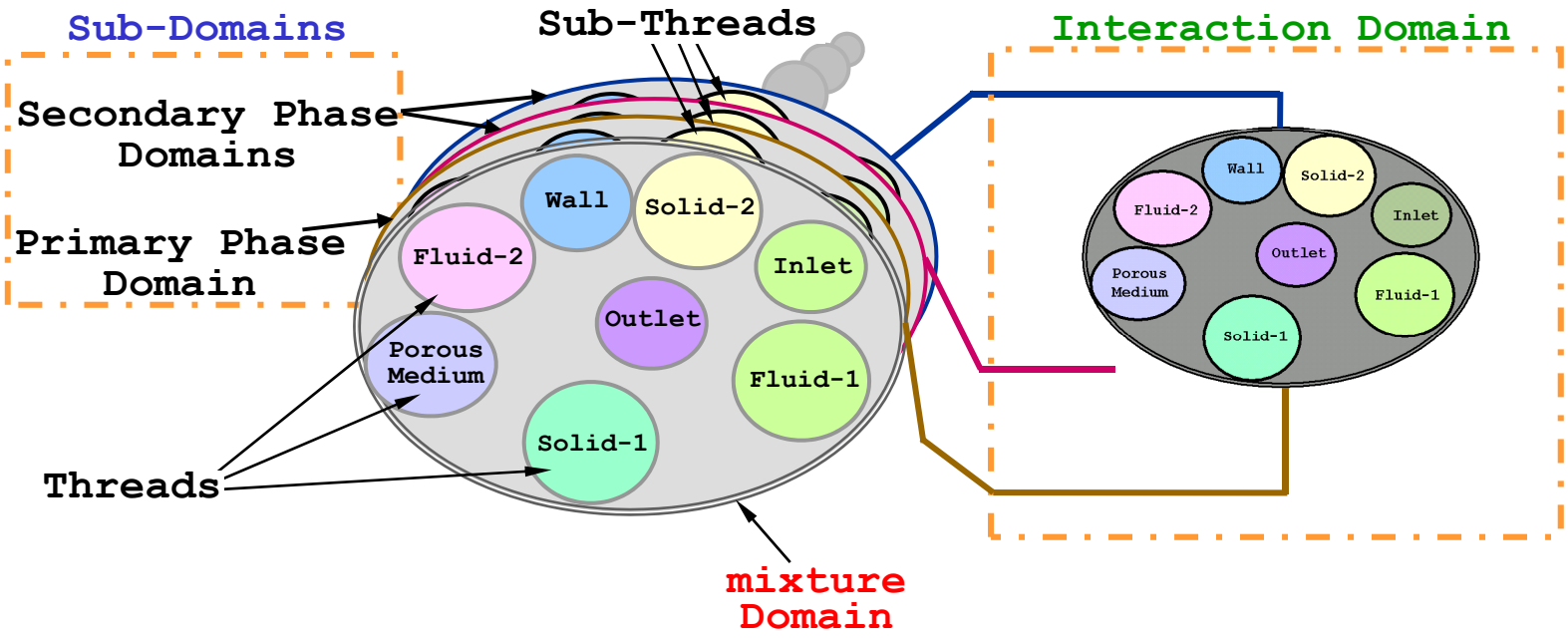
UDFs for Multiphase Flows

Advanced Fluent User Defined Functions



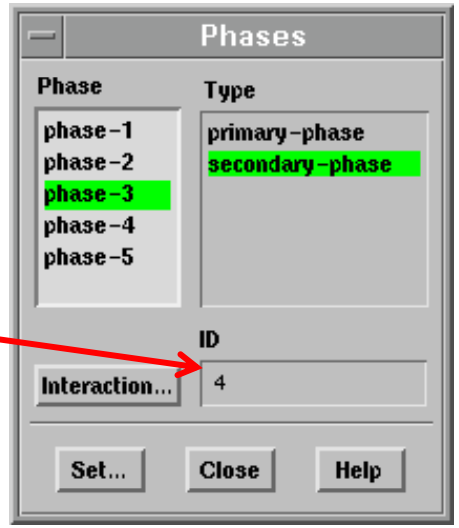
Additional Data Structure for Multi-phase

- **In multiphase models there are multiple domains involved:**
  - **Mixture (Super) Domain: This is the top-level domain contains all phase-independent and mixture data: geometry, connectivity, property**
  - **Sub-domains (phase domains): Each phase has a sub-domain that inherits the mixture-specific data and maintains the phase-specific data**
  - **Interaction Domain: To activate the phase interaction mechanisms**



Domain Pointers and Phase Domain IDs

- **For multiphase models, the domains need to be identified by unique domain IDs (including the interaction domain)**
- **`Domain_ID` of the *mixture domain* is always 1, but values for phase domains (primary and secondary) can be any integer greater than 1. The values can be found in the FLUENT GUI panel under define>phases**
- **The macro `GET_Domain(domain_id)` returns either the mixture, a phase or an interaction domain pointer. (But the mixture domain pointer is always obtainable by using Get_Domain(1)!)**
- **In comparison, only one domain (ID=1) exists for single phase flows**



Phase Domain Index

- **Phase Domain Index (PDI) is an integer index for a given subdomain (not to be confused with the domain ID, see an example below):**
  - **phase_domain_index=0 for the primary phase**
  - **1 for the first secondary phase, …, N-1 for the last secondary phase**
- **Phase domain index is primarily used to retrieve phase domain pointers (and subthread pointers via different macros)**
  - **Especially when you want to access data for a particular phase:**

```
Domain *subdomain=DOMAIN_SUB_DOMAIN(mixture_domain,
        phase_domain_index);
```

  - **The cell thread pointer in the subdomain gives the phase data (e.g., thread_loop_c (t, subdomain) loops over all cell threads of the phase)**
  - **The inverse macro to get PDI is:**

```
int phase_domain_index=PHASE_DOMAIN_INDEX(subdomain);
```

- **For example, a problem using the VOF model with 2 phases:**
  - **The mixture domain ID is always 1**
  - **Primary Phase: the phase domain ID is 2 ; the phase domain index =0**
  - **Secondary Phase: the phase domain ID is 3; the phase domain index is 1**

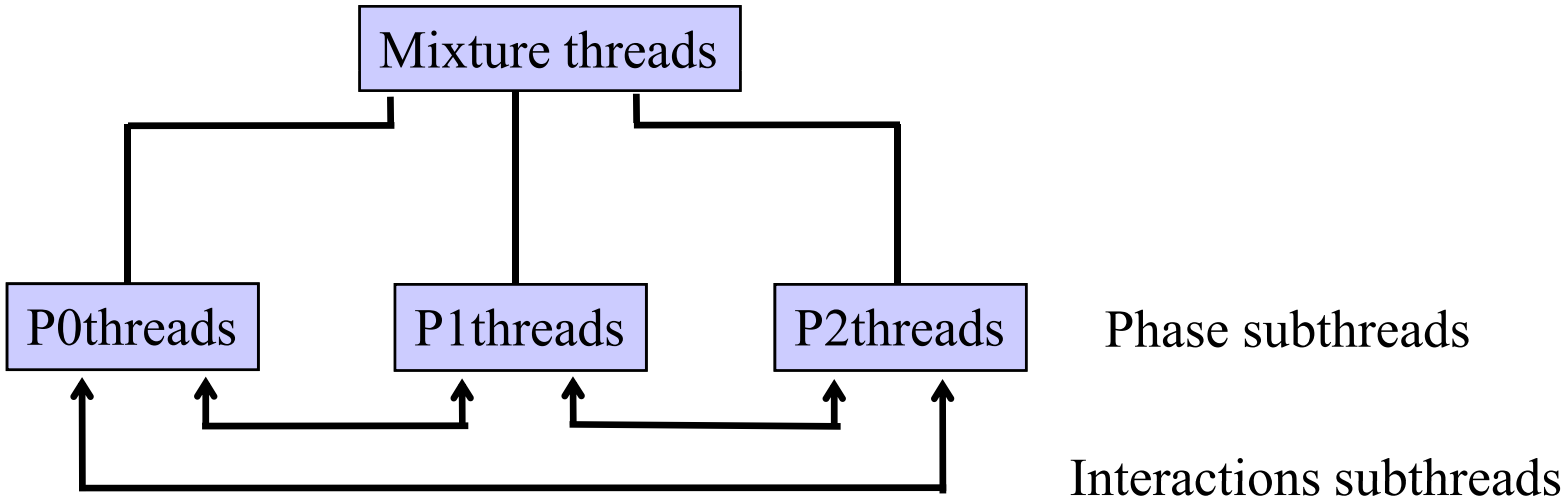
Accessing a Specific Phase Domain

- **Several macros are available for accessing a specific phase domain (or sub-domain) pointer:**
- `Domain *d=Get_Domain(domain_id);`
  - **For example, domain_id=1 for the mixture domain, 2 for the primary phase, 3 for a secondary phase, and so on**
- `Domain *subdomain=DOMAIN_SUB_DOMAIN(mixture_domain, phase_domain_index);`
  - **For example, phase_domain_index=0 for the primary phase, 1 for the secondary phase, etc.**
- **Once the domain pointer is available, loop over cell threads, and over all cells within each cell thread as follows:**

```
thread_loop_c(cellthread_ptr, subdomain)
  {
    begin_c_loop(c, cellthread_ptr)
      {       }
    end_c_loop(c, cellthread_ptr)
  }
```

Mixture Threads and Phase Subthreads

- Threads possess a similar hierarchy that matches that of the domains
- The **"mixture threads"** are where the "mixture" of the phases is stored and so are often called the **"superthreads"**
- Phase-independent data such as the cell's geometry data are stored in the mixture thread
- Each Phase has its own set of threads known as a **"subthreads"** or "**phase threads**"



Retrieve the Mixture Thread

- **Mixture thread pointer is retrievable if a particular phase-level pointer (subthread) is available:**

```
Thread *subthread;
Thread *mixture_thread=THREAD_SUPER_THREAD(subthread);
```

- **Mixture (super) domain pointer is retrieved, then `Lookup_Thread` macro is used to get the mixture_thread pointer (the particular cell zone ID must be supplied):**

```
Domain *mix_domain=Get_Domain(1);
subthread=Lookup_Thread(mix_domain, ID)
```

- **Using `thread_loop_c (mixture_thread, mix_domain)` is also an option**

Accessing a Specific Subthread

- **We can also access phase threads (subthreads) by a number of macros:**
  - `thread_loop_c(mixture_thread, mixture_domain)` **Once a subthread pointer is available, we can retrieve information through macros like** `begin_c_loop` **or** `begin_f_loop`
  - `THREAD_SUB_THREAD`

```
int p_d_index = 0; /* primary phase domain index is 0 */
Thread *mix_thread;/* mixture-level thread pointer */
Thread *subthread=THREAD_SUB_THREAD(mix_thread, p_d_index);
```

Given a `ph_d_index`**, THREAD_SU_THREAD returns the corresponding phase-level sub-thread pointer**

  - `THREAD_SUB_THREADS`

```
Thread *mixture_thread;
Thread **pt=THREAD_SUB_THREADS(mixture_thread);
```

Given a pointer to the mixture thread, returns a phase thread pointer array: pt[j] is the subthread pointer to the j-th phase (i.e., phase_domain_index=j)

- **Note that the mixture_thread pointer(s) are assumed to be available. If not, use one of the methods discussed in the previous slide**

Loop over Phase Domains in a Mixture

- **Loop macro for visiting all sub-domains (primary and secondary phases) in the mixture:**

```
sub_domain_loop(sub_domain, mixture_domain,
     phase_domain_index)
```

- **Example: Given a `mixture_domain` pointer, the following macro loops over all phases in the mixture. `sub_domain` and `phase_domain_index` vary within the macro**

```
sub_domain_loop(sub_domain, mixture_domain, p_d_i)
{
  .........
}
```

Additional Loop Macros in Multiphase

- **Still more loop macros are available. Not only they loop through the cell or face threads, they also returns a pointer array containing the all the phase-level thread pointers, which are the required arguments in accessing phase-level data. For cell data, we can use the following:**

```
mp_thread_loop_c(mixture_thread, mixture_domain, pt)
```

  - `mixture_thread` **is the pointer to mixture thread in the** `mixture_domain`
  - `mixture_domain` **is assumed to be available (if not, use Get_Domain(1) !)**
  - `pt` **is an array of phase-level thread pointers: the pointer to cell thread for the j-th phase is** `pt[j]`, **and** `j` **is the** `phase_domain_index (PDI)`:

```
0: primary phase
1, 2, …: secondary phase(s)
```

  - **Inside the loop, any phase data can be accessed easily. For example,** `C_R (c,pt[0])` **for primary-phase density,** `C_R (c,pt[1])` **for the first secondary phase density, and** `C_R(c,mixture_thread)` **for the mixture density**

Additional Loop Macros in Multiphase

- **For face zones, we can use the following:**

  `mp_thread_loop_f(face_threads, mixture_domain, pt)`
  - `face_threads` **is a pointer to face thread in the** `mixture_domain`
  - `mixture_domain` **is assumed to be available**
  - `pt` **is an array of phase-level face thread pointers: the pointer to face thread for the j-th phase is** `pt[j]`**, where** `j` **is the** `phase_domain_index`

- `sub_thread_loop` **macro loops over all phase-level threads associated with a mixture_thread**

```
int phase_domain_index;
Thread *subthread, *mixture_thread;
sub_thread_loop( subthread, mixture_thread,
                 phase_domain_index)
```

Summary

- **The most important concepts in UDFs for multiphase flows have been covered in this lecture**
- **It is important to know how to retrieve the "mixture threads" (superthreads) or a specific "subthreads" when accessing the flow/fluid data**
- **The same cell index (cell_t c) is used by all different threads in any loops**
- **FLUENT's UDF Manual provides documentation for macros and good examples**