

Numeric Algorithms

Solutions

- What does `iota()` do?
 - `iota()` populates a container with elements whose values successively increase by 1
- What arguments does `iota()` take?
 - Iterator range and the value of the first element
- Write a simple program that uses `iota()`
- Write the equivalent program without using `iota()`

- What does `accumulate()` do?
 - `accumulate()` returns the sum of all the elements in an iterator range
- What arguments does `accumulate()` take?
 - The iterator range and the starting value of the sum
 - Optionally, a function which is used to perform the addition
- Write a simple program that uses `accumulate()`
- Write the equivalent program without using `accumulate()`

- Write a simple program that uses `accumulate()` and a user-provided function. Implement the user-provided function as
 - A lambda function
 - A functor

- (Optional) Describe what the following algorithms do. Suggest an situation in numerical computation where they could be applied:
 - `partial_sum()`
 - Populates an iterator range with the running sum of elements from another container $\{a_1, a_1 + a_2, a_1 + a_2 + a_3, \dots\}$
 - `adjacent_difference()`
 - Populates an iterator range with the difference of successive elements from another container $\{a_1, a_2 - a_1, a_3 - a_2, \dots\}$

- (Optional) Describe what the following algorithms do. Suggest an situation in numerical computation where they could be applied:
 - `inner_product()`
 - Multiplies the corresponding elements from two iterator ranges together, then calculates their sum
 - As usual, we can override the default `+` and `*` operators
 - Can be used in many situations where there is a pairwise operation followed by an aggregate operation, e.g. finding the largest discrepancy in data samples