GENERATING THE MINIMUM VALUES FOR THE ROTOR POTENTIAL

Solving the issue of *SEGFAULT* on compilation time

While generating the stack which holds the values of the triaxial potential $V(q)$, where q takes values in a definite interval, some issues with regard to the validity of the actual values need to be pointed out.
As a general condition, for every value of θ, the V stack. (denoted hereafter with `Vstack`) is constructed by using the given analytical expression, which depends on $V = f(\mathrm{X}, \theta, q)$. Because θ and X are fixed, the only variable is the q - coordinate which enters in the calculus of the Jacobi elliptic functions.
The parameter x goes inside a definite interval but with small steps, and at each step is generating a value $V(q)$.

The present algorithm is constructed in such a way that two possible outcomes are available to the runtime `Vstack`:

- get the current potential value at q.
  1. if the value of the potential real, then it is inserted into an array (using `emplace_back`)
  2. if the value is non-physical (actually equal to `6969`, which is a safety measure inside the code for dealing with non-physical solutions)

When the algorithm is iterating through the values of θ, each `Vstack` is created, then a class object is instantiated inside the potential method, where this object will take the potential stack as a parameter (actually as an `rvalue` reference) and then immediately return its smallest value and the index of it.

```
class Minimum
{
...
};
Minimum Object(vstack);
//the Object takes the stack and return its smalles value.
```

In the class implementation, there was a specific feature where the pair of (smalles,greatest) elements of the array would be automatically generated via the `std::minmax_element` function. This pair is stored with the help of an `auto` variable, then it is dereferenced:

```
auto minmaxPair = std::minmax_element(first, last);
auto minvalue = *minmaxPair.first;
```

Main issue

If the incoming potential stack (the `source` array) is empty, such a dereference is not possible, since the first and last (which are iterators) cannot work properly for an empty array. ***SEGFAULT*** was produces since `minvalue` wanted to take the value of a pointer which doesn't exist in the first place.

How to solve this?

Juts make sure the array is not empty. In the same method that does the evaluation of the minimum element, create an `if`-statement which increases the size of the array by 1-one element: using the same `emplace_back`.