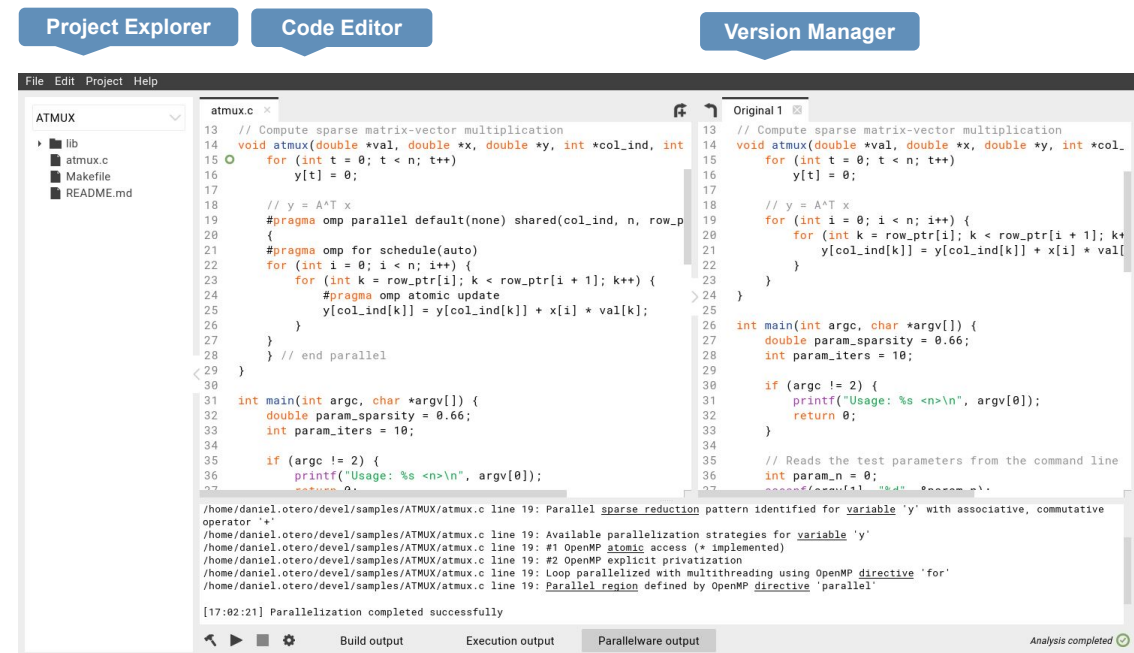


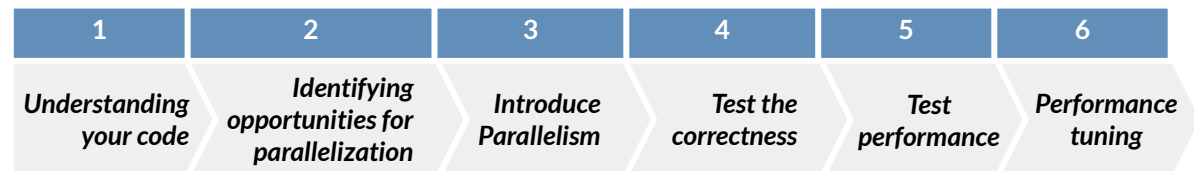
# Parallelware Trainer

## Getting started with Parallelware Trainer

1. **Launch the installer** (available for Linux, MacOS and Windows).
2. **Download a trial license:**  
<https://appentra.com/products/parallelware-trainer/trial/>
3. **Launch Parallelware Trainer and activate the licence:**  
Follow the instructions to update your licence.
4. **Open a project** using File->Project and navigate to the top level folder/directory of your software project
5. **Configure your project:**  
To build, run and analyze, go to Project->Configuration and enter:
  - **Analyze:** provide flags that you use in compilation to aid in the analysis of your code by Parallelware.
  - **Build:** the command for compiling your project (e.g. `make`).
  - **Run:** the command for running your project (e.g. `./my_code`).Click on Advanced to set variables such as `OMP_NUM_THREADS`.



www.appentra.com  
info@appentra.com



## The Parallelware software development life-cycle

### 1. Understanding your code

- a. Load your code by double clicking the required file in the Project Explorer.
- b. Use the green circles to help identify areas suitable for parallelization.
- c. Look for warnings to make your code follow best practices.

### 2. Identifying opportunities for parallelization

Click the green circles next to functions that correspond with your computational hotspots. Select your desired parallelization method from the dialogue.

### 3. Introduce Parallelism

- a. Closing the dialogue produces source-code with directives.
- b. Use the Parallelware console to identify other options.

### 4. Test the correctness

- a. of the parallel implementation (run debug tests).
- b. Look for defects (i.e. race conditions).

### 5. Test performance

of the parallel implementation (run benchmarking tests).

### 6. Performance tuning

- Iteratively test new options, ensuring you complete step 4 and 5 each time.
- a. Use the suggestions in the console for other parallelization options.
  - b. Re-check for new hotspots regularly.
  - c. Keep testing correctness and performance.