Altair Compose Quick Start Guide for IIT
Bombay Techfest Altair Contest

Altair Compose provides engineers, scientists and product creators a high level, multi-language programming environment which enables you to efficiently perform numerical computations, develop algorithms, analyze & visualize various types of data. It supports multiple languages such as Tcl or Python and offers OML (Open Matrix Language) a matrix-based numerical computing language for all types of math from solving matrix analysis, differential equations to performing signal analysis and control design.

Interfacing with Other Languages

Compose supports OML, Tcl and Python languages. The use of the interpreter language is first defined by the script being executed.

Command Windows for either OML or TCL or Python file languages can be displayed and commands can be entered and executed directly.

Overall Altair Compose is a language, it’s also programming environment and provides with whole bunches of math library along with visualization.

The Environment

Altair Compose is an Integrated Development Environment (IDE). The below Figure shows the Altair Compose in authoring mode. The description of various toolbars in the default authoring mode is described below.
Basic concepts

Variable Initialization

To store a value in a program or in any session in Altair Compose, a variable is used. The Variable window shows variables that have been created and their values.

name_of_variable = value or expression

For example:

“a = 4”, “b = 6” and “c = 345” as shown below figure.

\[
\begin{align*}
4 & \quad a = 4; \\
5 & \quad b = 6; \\
6 & \quad c = 345;
\end{align*}
\]

Initialing of variable

Run the script and type “c” in the Command Window to confirm the value of “c”.

Use of Semicolon (;)

A semicolon (;) indicates the end of a statement. Whenever you want to suppress and hide the output for an expression, you can add a semicolon after the expression.

```
> x=2;
> y = 3 + 2
y = 5
```

Semicolon used in Altair Compose

Adding Comments

The percent symbol (%) is used for indicating a comment line. A comment is not executed when the script is run. For example:

```
x = 2 % Defining x and initialising it with a value
```

Simple Mathematical Operations

Altair Compose can be used to perform simple mathematical operations such as addition, subtraction, multiplication and divisions.

```
> 2+4
ans = 6
```

Simple addition example

In this case “ans” is the last answer of compose which is the result of the addition. As an example, in the command window type “a =9” and “b = 3” and do the following operation as shown below.
Operators in Altair compose

Using Compose as a calculator, it respects the rule of multiplication and division first as well as parentheses rules, like the following example shows:

Simple multiplication and division example

You can also store values into variables for later calculation as the following example shows:

Simple variable example

Besides addition, subtraction, multiplication and division, you can also use exponentiate and root numbers and variables. For exponentiating simply use the “^” operator and for the square root, you can use “sqrt(x)”. 
Matrices Input

For many types of math operations, matrices and tensors are the way to organize your values. You can define a matrix with multiple rows and columns in Altair Compose with the following input format:

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{bmatrix}
\rightarrow \begin{bmatrix}
1,2,3;4,5,6;7,8,9
\end{bmatrix}
\text{or}\ [1\ 2\ 3;\ 4\ 5\ 6;\ 7\ 8\ 9]
\]

So, the “,” or “space” operator separates columns and the “;” operator separates rows, like the following Figure shows:

Matrix Input in Altair Compose

Basic commands

“clear”
This command clears all variables from memory. Example is given below.
Altair Compose “clear” command

“clc”

This command empties out the display of the command window. Example is given below.

```
> a = 1
a = 1
> a
a = 1
> clear
> a
Unknown function: a at line number 1
```

Altair Compose “clc” command

“disp”

disp(x) - Prints the value of x to the application’s Command window.

```
> disp('This is a tutorial')
This is a tutorial
```

Altair Compose “disp” command

In Altair Compose, literal strings are denoted using the single quote (next to the Enter button on typical keyboard).

“for”

The “for” loop is supported in Altair Compose using the syntax and provides the ability to define an incremental loop. Example is illustrated below.
The increment in "for" loop doesn’t have to be an integer. Example is illustrated below

```
> total = 0;
> for i = 1:10
  total = total + i
end;
> total = 1
> total = 3
> total = 6
> total = 10
> total = 15
> total = 21
> total = 28
> total = 36
> total = 45
> total = 55
```

**Help Options**

Altair Compose comes with a knowledge base where you can search, work through tutorials and more. The easiest way to access it is by pressing F1 on your keyboard. Your web browser will open and you can access the help files and search for commands and/or tutorials.
When you search for a new command, we recommend using a few words, since commands might be used in different contexts than you might expect. Sometimes, it can be useful to use synonyms if the search is not successful at the first attempt.

We can select objects with the right mouse button which gives help access to that object from within the editor or command window. It also contains other options depending on the object and window. See figure below
Altair Compose options on help.
General help can be accessed through the File > Help menu:

![Altair Compose General Help Window](image)

**Videos**


1. Write a program
- To find addition, subtraction, multiplication and division of two numbers.
- To exchange the values of two variables without using the third variable.

Compose code
```
clc; clear all;
a = input('enter a value = ');
b = input('enter b value = ');

% simple math operation
add = a+b
sub = a-b
mul = a*b
div = a/b

% swapping a and b variable values without using the third variable
a = a+b;
b = a-b
a = a-b
```

- Run compose
- This launches the Altair Compose MATH GUI in the Authoring Mode and by default a file named Untitled1.oml exists in the Editor Window.
• Under tab “View” tick the features we want to enable. Here the most common ones:

3. Move to the desired directory through the file browser of Compose
4. Write an .oml script, in the Editor.

5. Save the script .oml in the working directory by selecting single folder icon as shown.

6. run .oml (clicking on the Start button in the Editor toolbar)

7. We can see the answer in command window.

2. Find the area of a circle with diameter of 10 cm.
   \[ A = \pi r^2 \]

   Compose code

   ```matlab
   clc; clear all;
   d = 10; % diameters(cm);
   r = (1/2)*d; % radius;
   area_of_circle = pi*r*r % cm2
   answer:
   area_circle = 78.5398163
   ```