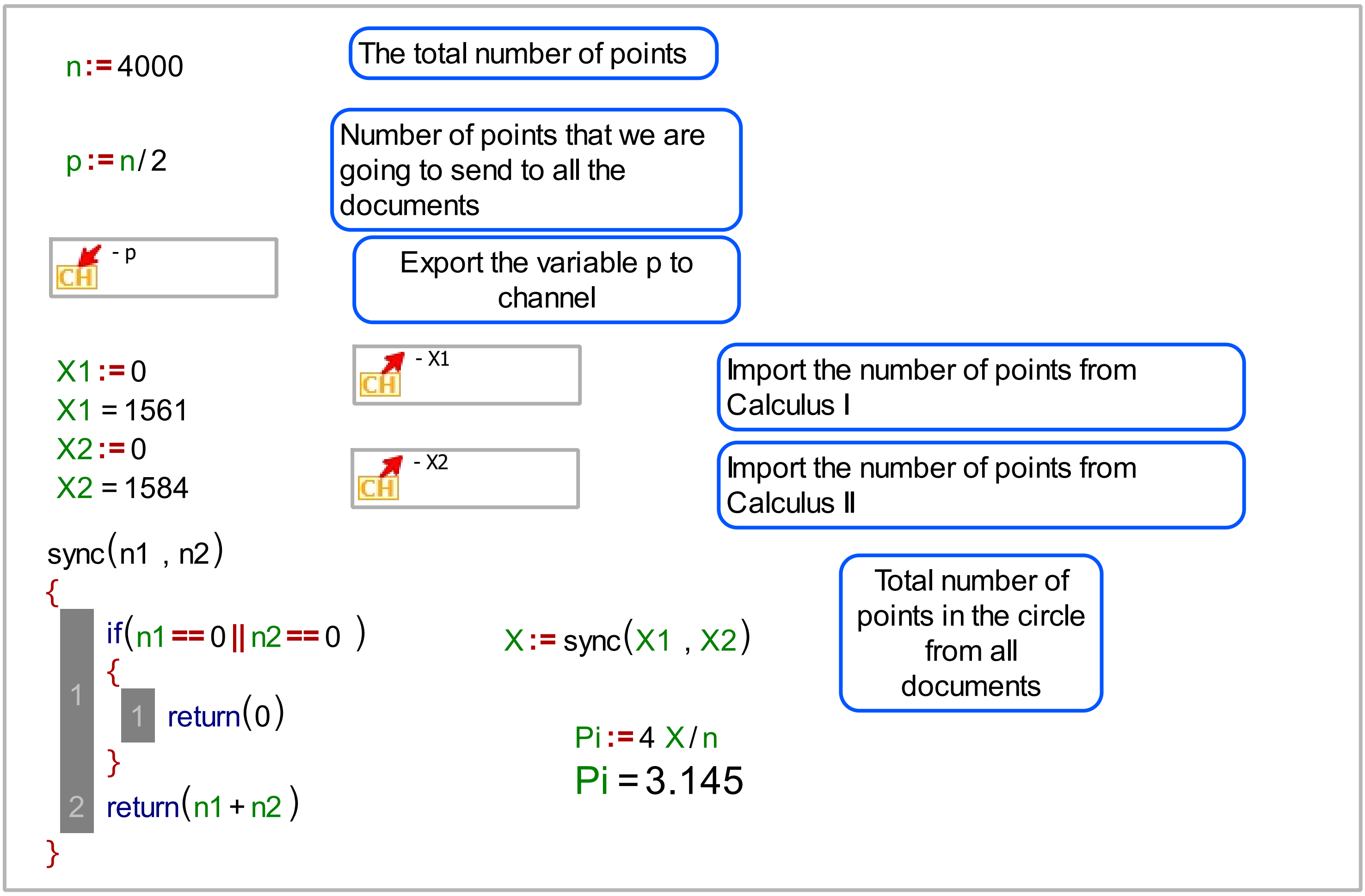
Pi Calculation
Parallel Processing

Main document

This is the master document in the parallel processing example. We derived the problem of Pi number approximation in the documents "Parallel processing 1.mdd" and "Parallel processing 2.mdd." They are both working calculations and the results are collected in this Main Document. The basic idea of how to calculate the Pi approximation is described in "Parallel processing - graph.mdd".

The algorithm of Pi number approximation, which we are trying to demonstrate, is as follows:

- We have created the variable, n, with the value that represents the total number of points that we are going to create randomly;
- We have then divided this number in two equal parts (variable p in the document) and exported it through the channel in-between each of the documents;
- Each of the mentioned work documents will import the value of variable p from the channel and randomly create that number of points by uniform distribution within (-r, r) interval and checks the number of points which are placed inside the circle which has a radius of r;
- Each document will export this number of calculated points into the channel;
- Finally, we import these two values from the channel, sum them up, and use them in the equation to approximate the Pi number.



```
n:=4000
```

The total number of points

```
p:=n/2
```

Number of points that we are going to send to all the documents

CH - p

Export the variable p to channel

```
X1:=0
X1 = 1561
X2:=0
X2 = 1584
```

CH - X1

Import the number of points from Calculus I

CH - X2

Import the number of points from Calculus II

```
sync(n1 , n2)
{
    if(n1==0||n2==0 )
    {
        return(0)
    }
    return(n1+n2 )
}
```

```
X:=sync(X1 , X2)
```

Total number of points in the circle from all documents

```
Pi:=4 X/n
Pi = 3.145
```

When we divide the process into two documents, the generating and examining of the points will be split between documents. By doing this the calculation will be done simultaneously on either document which makes it practically twice as fast. You can only imagine how fast and efficient the process will be if you divide the calculation on a greater number of computers. Furthermore if you compile these two files into EXE files (Build and Run Method), you can achieve much faster speeds.

To run this current system of files you must first open the main parallel processing document and then open the other two parallel processing files. Once you have done this, select the Math tab and then press the Evaluate icon.