Optimization tools for Radiance

KHALED NASSAR

MOHAMED ALY

THE AMERICAN UNIVERSITY IN CAIRO

الجامعة الأمريكية بالقاهرة

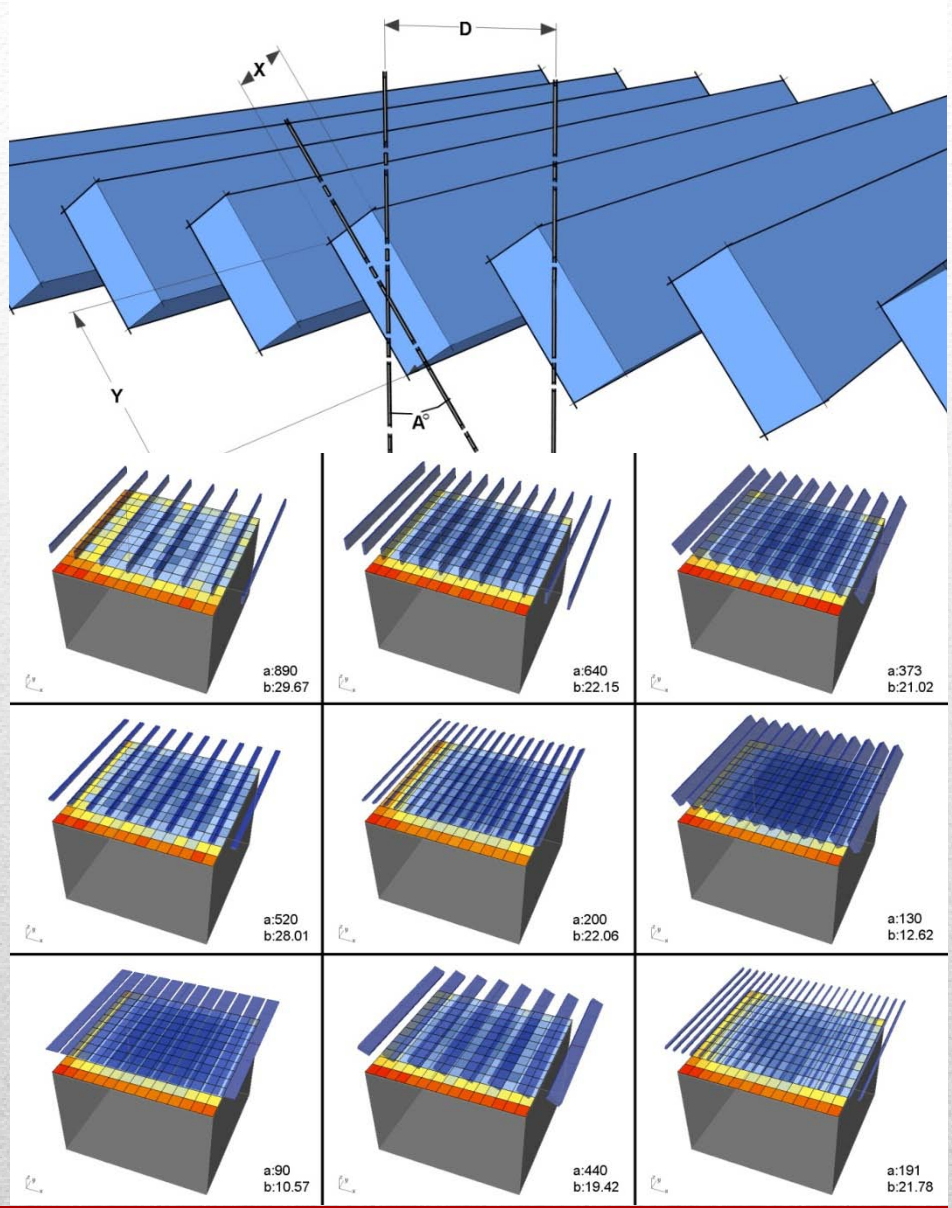
- Lets start with some examples of design optimizations in real world problems. I will present two problems

Why optimization

Problem: Design the roof shading to maximize the shading on the surface and at the same time provide for enough ventilation for the roof top equipment.

A Quick and “dirty” formulation was needed due to the time constraints for the project

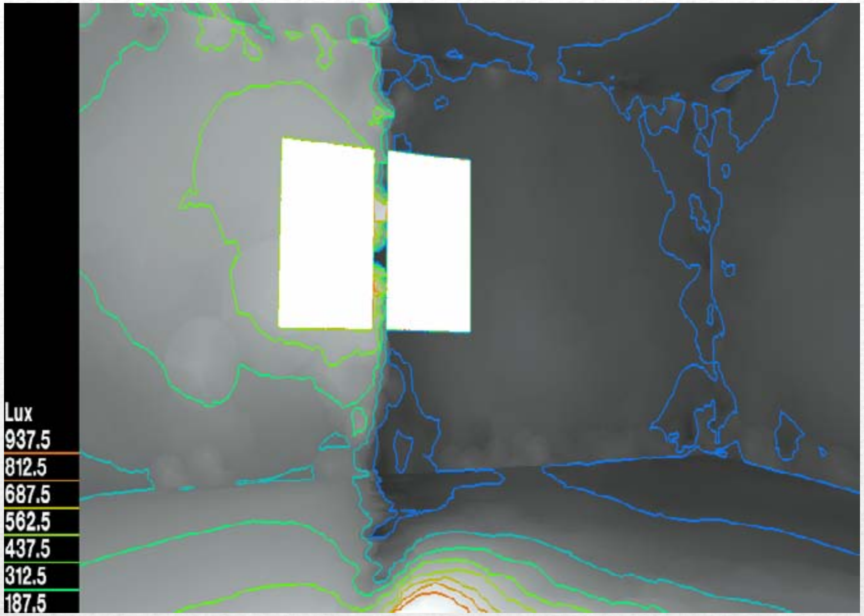
We try to find some design parameters related to the spacing, size, angle as even material of the sun shades. The objective will be a function with time and complexity required to solve the problem. One can even use rcontrib to calculate annual contribution of the sky/sun on the roof surface given a certain configuration

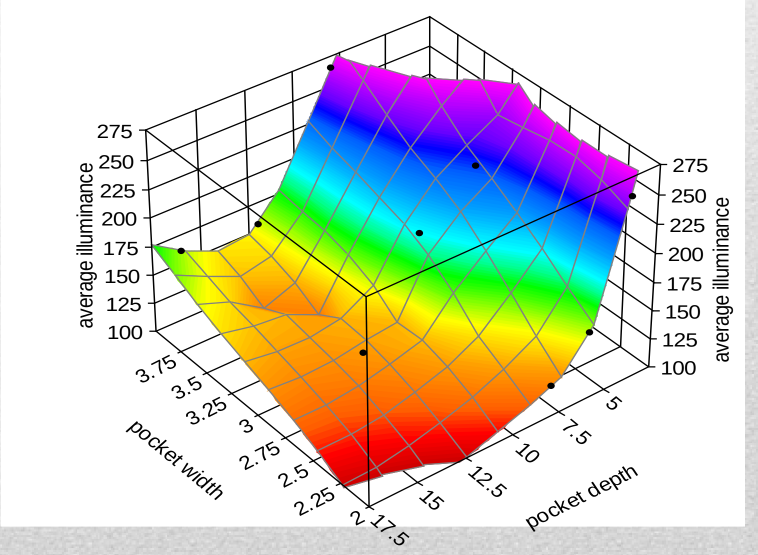


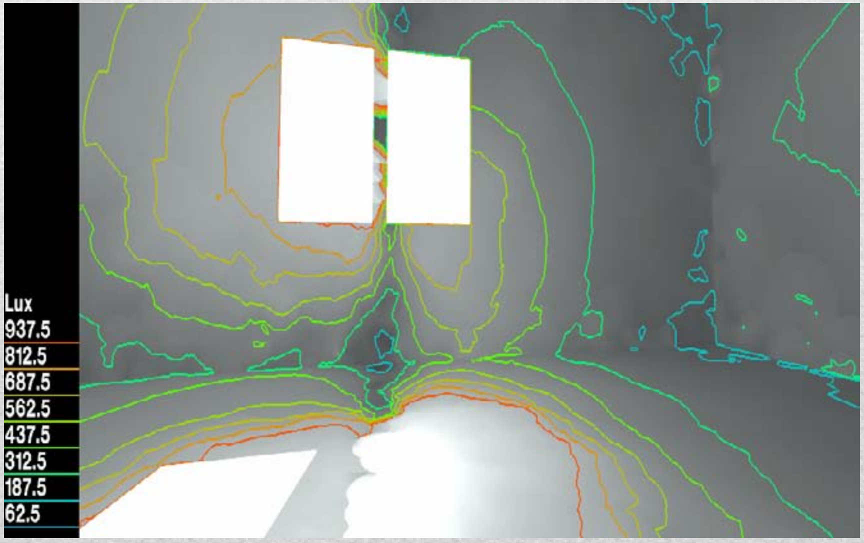
Problem 2 involves design a light pocket to maximize lighting levels on the inside rooms. You will be able to vary the dimensions of the pocket within constraints as well as the materials used.

Under clear sky conditions the solution is not always obvious

You could use trial and error but with just 5 variables each with 10 different options you end up with 1000s of solutions and the solution may not always be clear







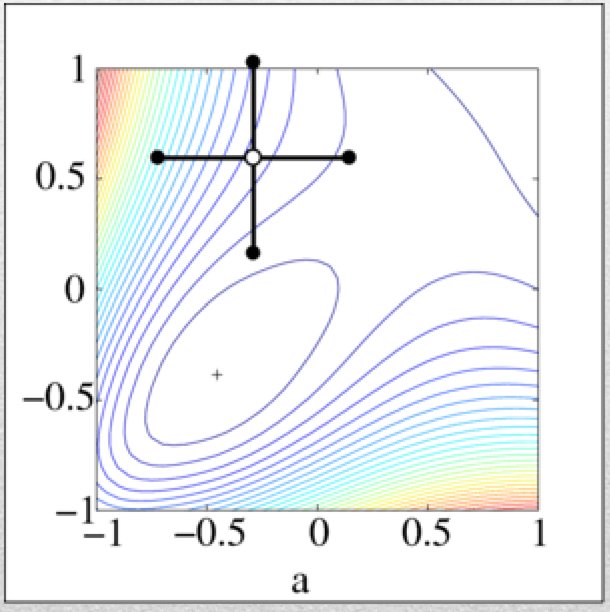
- This tutorial will demonstrate how to do optimization using radiance. It only assumes that you are familiar with radiance. Other software will be introduced and we will show how to use these tools to do optimization but other functions not related to the optimization will not be introduced.

What this tutorial is about

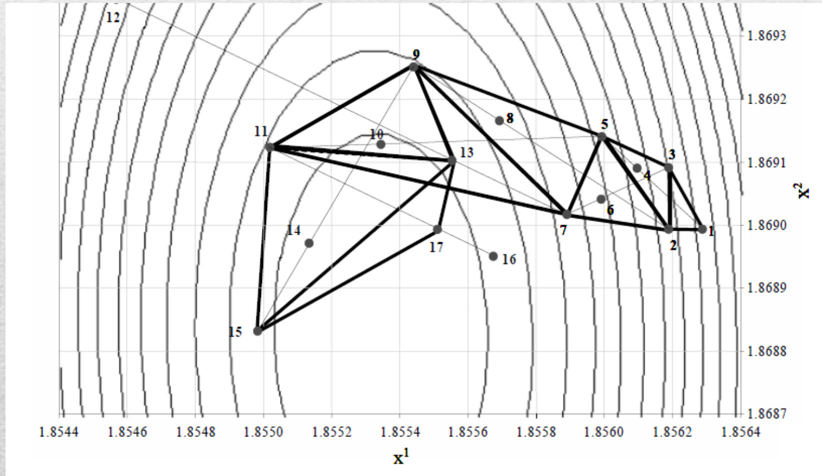
- Optimization algorithms for functions: y = x^2 +2x
- Optimizing functions without a closed form for the gradient such as the Radiance. Functions may be non-smooth, discont inuous and thus non-differentiable or we are not able to calc ulate a gradient because it is a result of a numerical method s uch as Monte-Carlo Sampling
- Evolutionary techniques and non gradient based techniques
- Overview of gradient-based techniques

Optimization Techniques

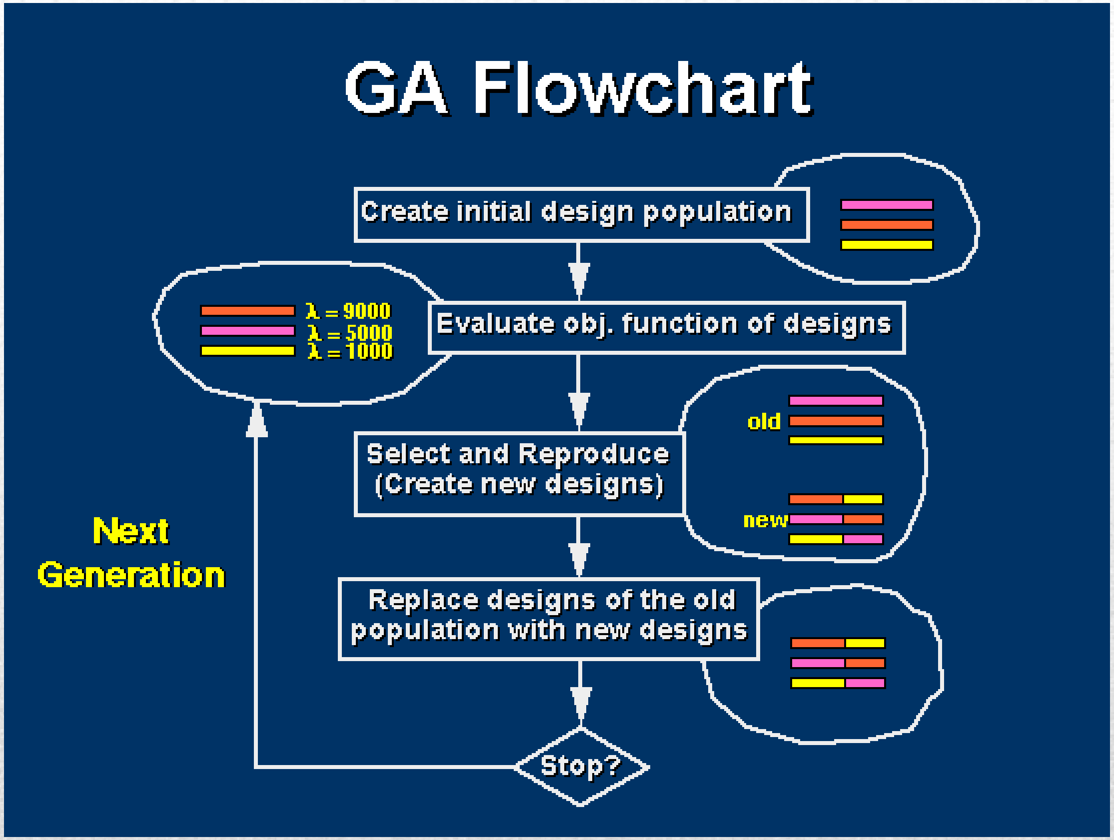
The Discrete Armijo Gradient algorithm approximates gradients by finite differences. Since the Discrete Armijo Gradient algorithm is sensitive to discontinuities in the cost function, LBL recommends not to use this algorithm if the simulation program contains adaptive solvers.



The Simplex algorithm constructs an n-dimensional simplex in the space that is spanned by the independent parameters. At each of the (n+1) vertices of the simplex, the value of the cost function is evaluated. In each iteration step, the point with the highest value of the cost function is replaced by another point. The algorithm consists of three main operations: (a) point reflection, (b) contraction of the simplex and (c) expansion of the simplex.



Gradient based techniques



Derivative Free. What **GPS** algorithms have in common is that they define the construction of a mesh, which is then explored according to some rules that differ among the various members of the family of GPS algorithms. If no decrease in cost is obtained on mesh points around the current iterate, then the distance between the mesh points is reduced, and the process is repeated.

Particle Swarm

Genetic Algorithms

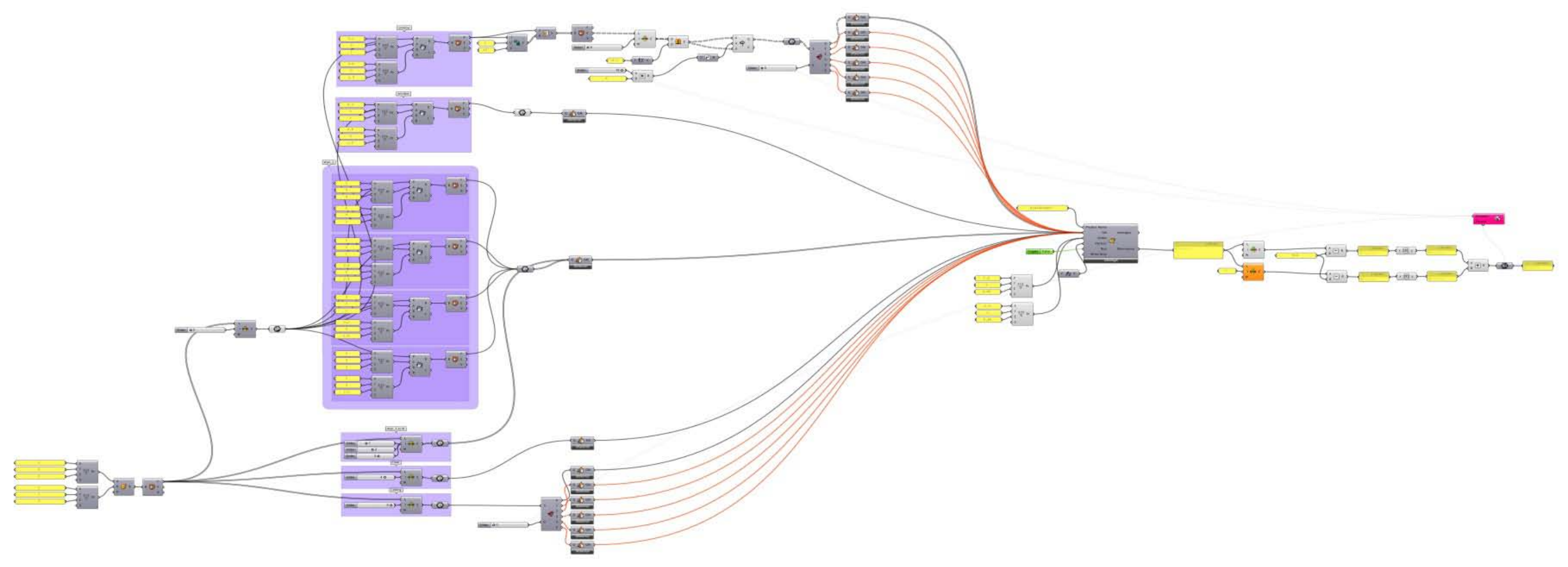
Evolutionary Techniques and non gradient techniques

- Two ways to do optimization using radiance
- We will compare between two techniques on the same example
- Using Rhino + Grasshopper + Galapagos
- The example is a simple room with a window and we would like to optimize the design of the room.
- We will divide the kinds of parameters we can optimize into two categories, geometric and non-geometric

Two ways to optimize using Radiance

ALTERNATIVE 1.

OPTIMIZATION USING DIVA RHINO GRASSHOPPER AND GALAPAGOS



Step 0. Makes sure you have grasshoper, diva for grasshoper and galapagos installed.

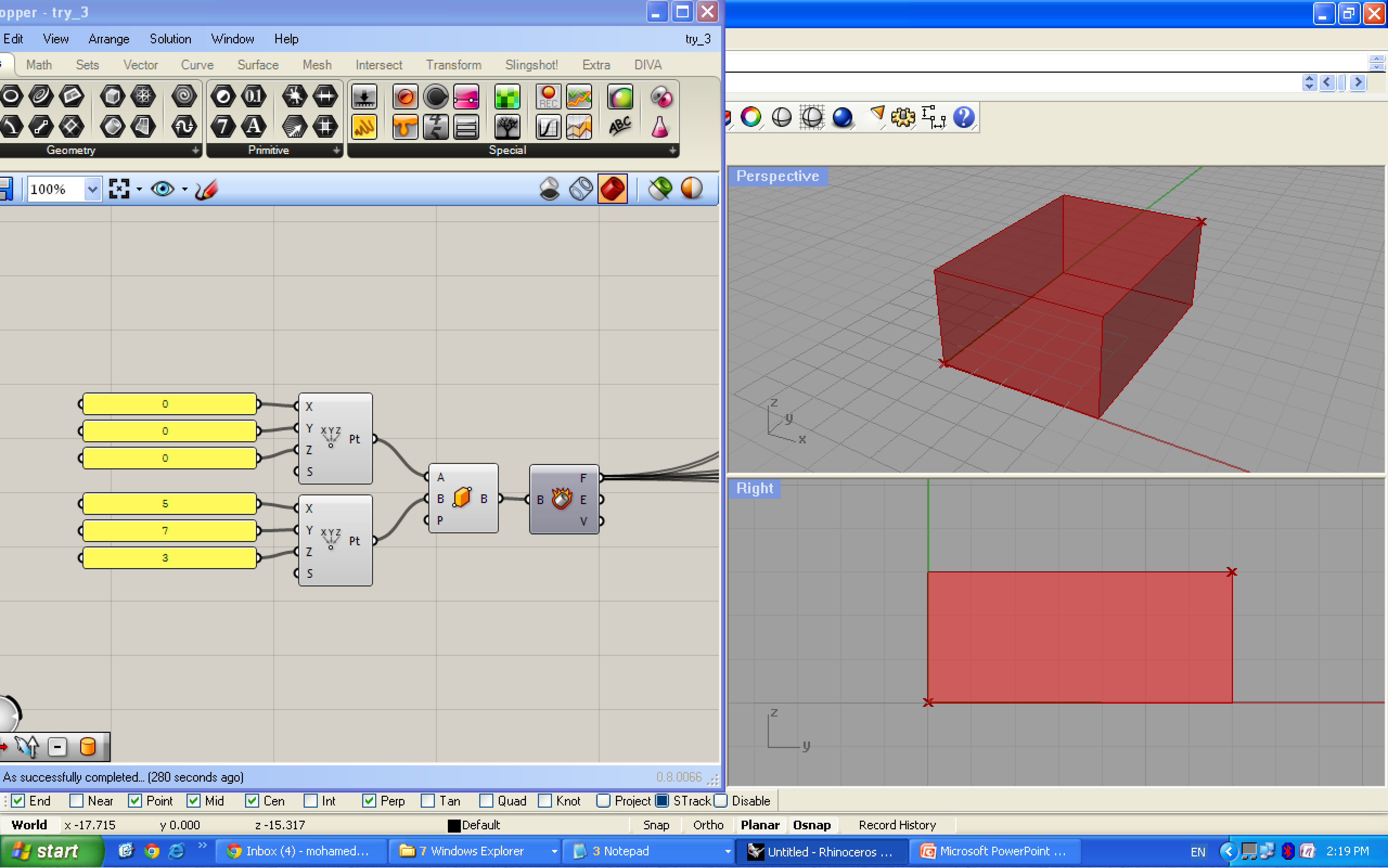
Step 1. Draw or import the design into Rhino. You can draw the room in rhino by...if you are using diva for rhino you can only do single analysis and not iterative or algorithmic design optimization. Therefore you need to use diva cor grasshoper. You will draw the same room algorithmiclly using grasshoper and the link it to radiance using diva for grasshopper.

To draw the room algorithmically using grasshoper:

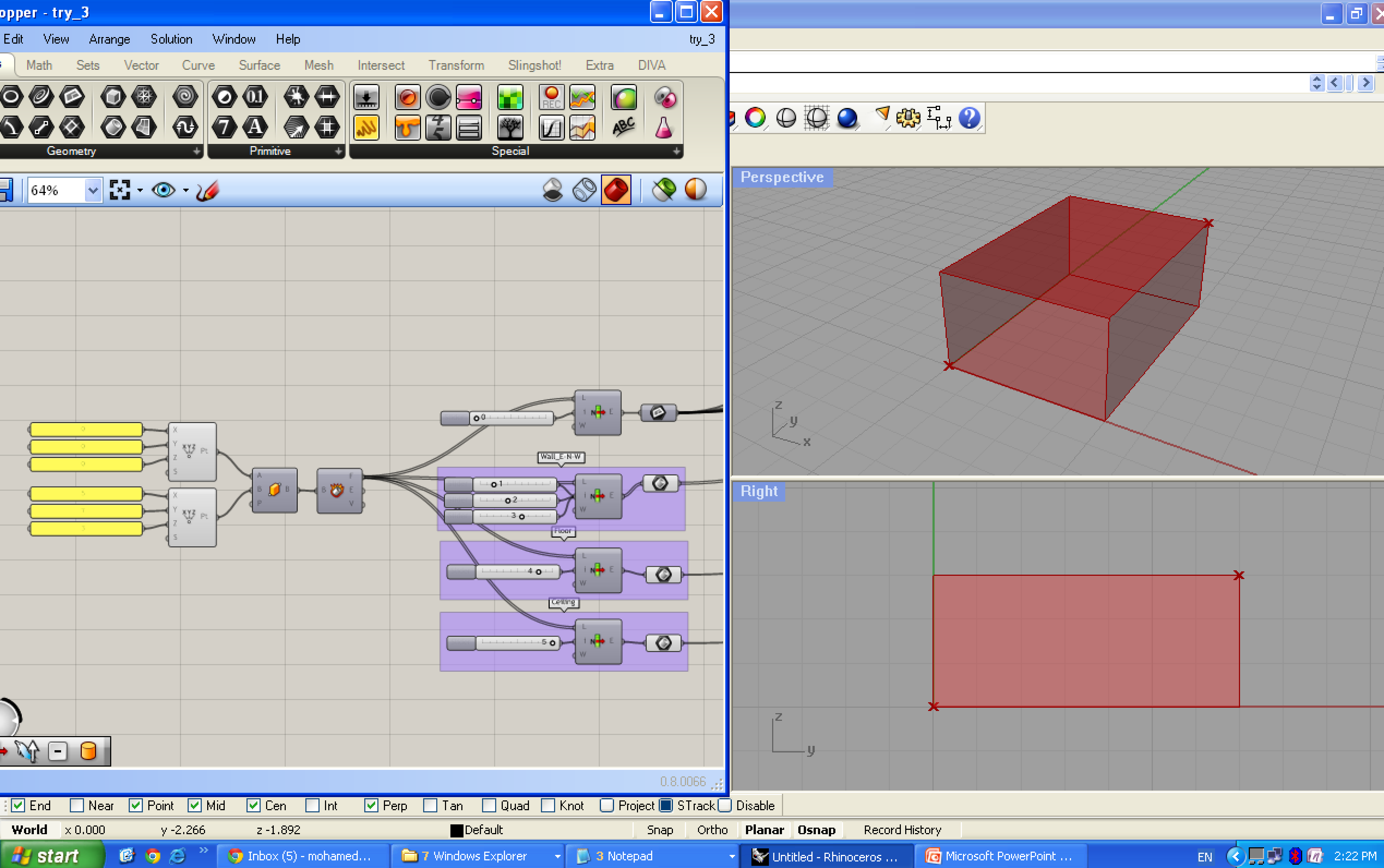
Step 2. Now we can change the various geometric parameters in grasshoper the use diva for grasshoper to link to radiance and we are able to vary the sliders related to these variables in order to observe the effect on the radiance analysis output. In order to do optimization you need to link this grasshoper definition to an optimization tool which is galapagos. So in this step we will a. use the parameter sliders as inputs to galapagos, b. define an objective function, and c. set optimization parameters

Step 3. Running the optimization and vary the parameters to try and get a better result

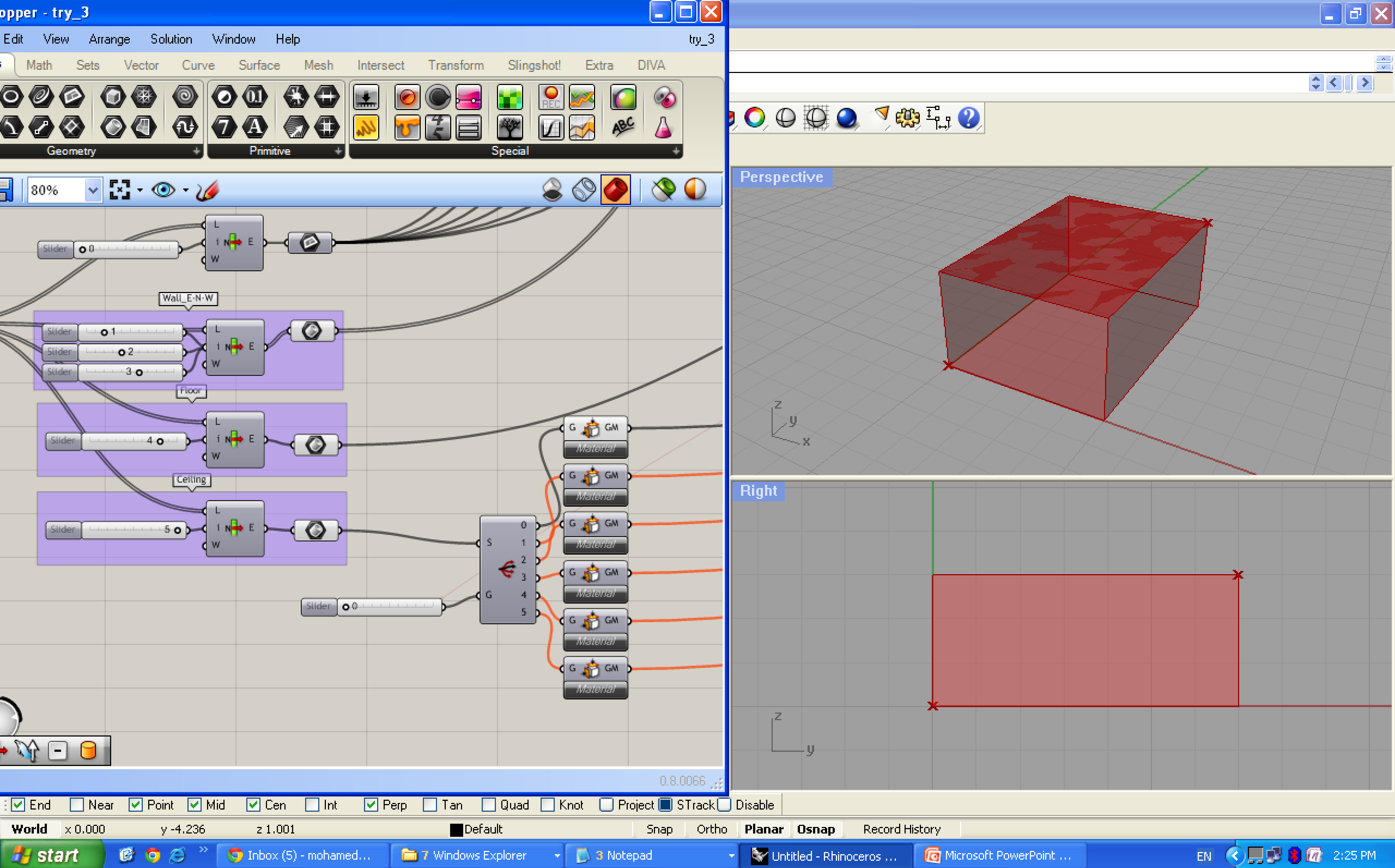
Steps Required



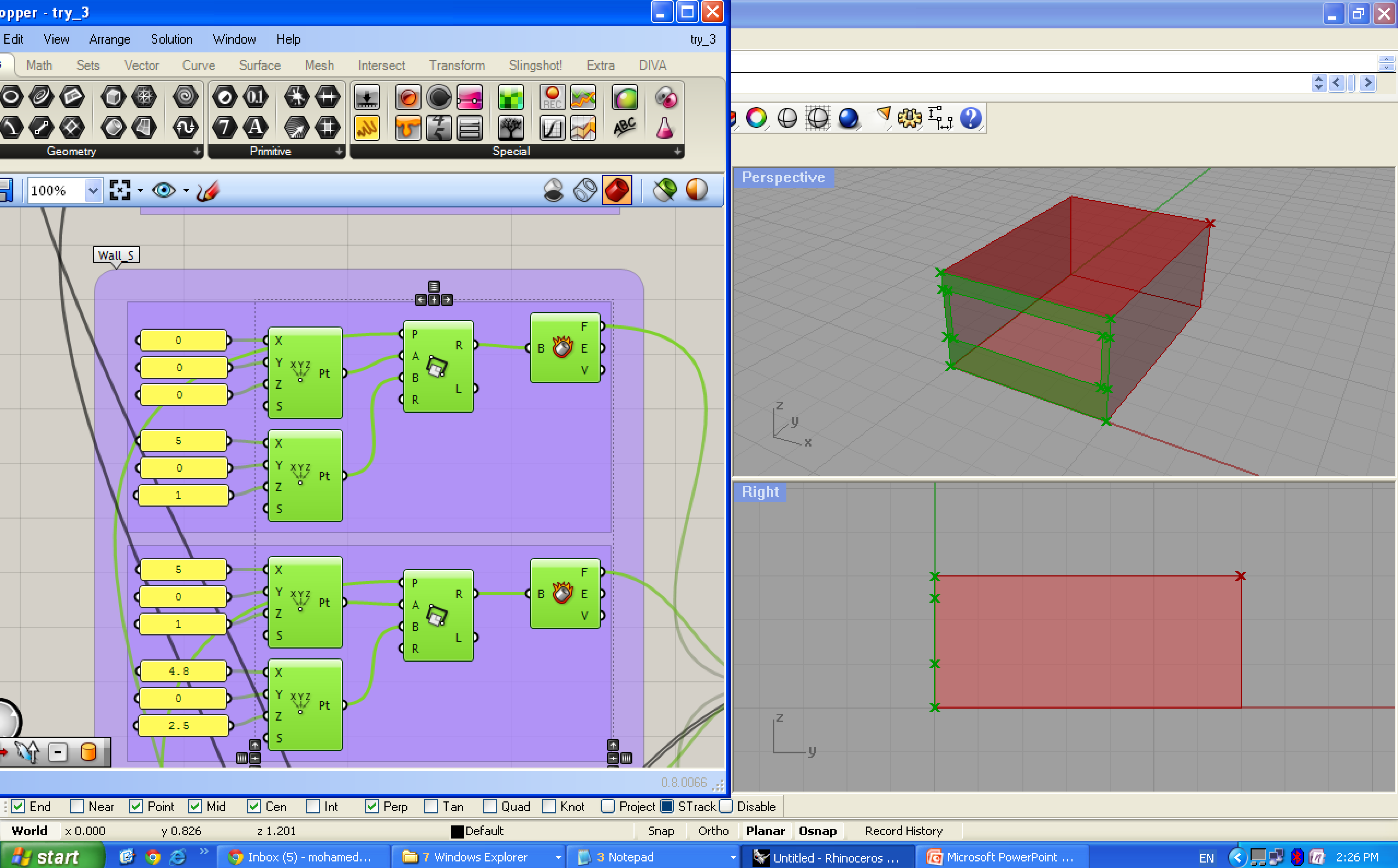
Creating a box using Two points, representing our tested



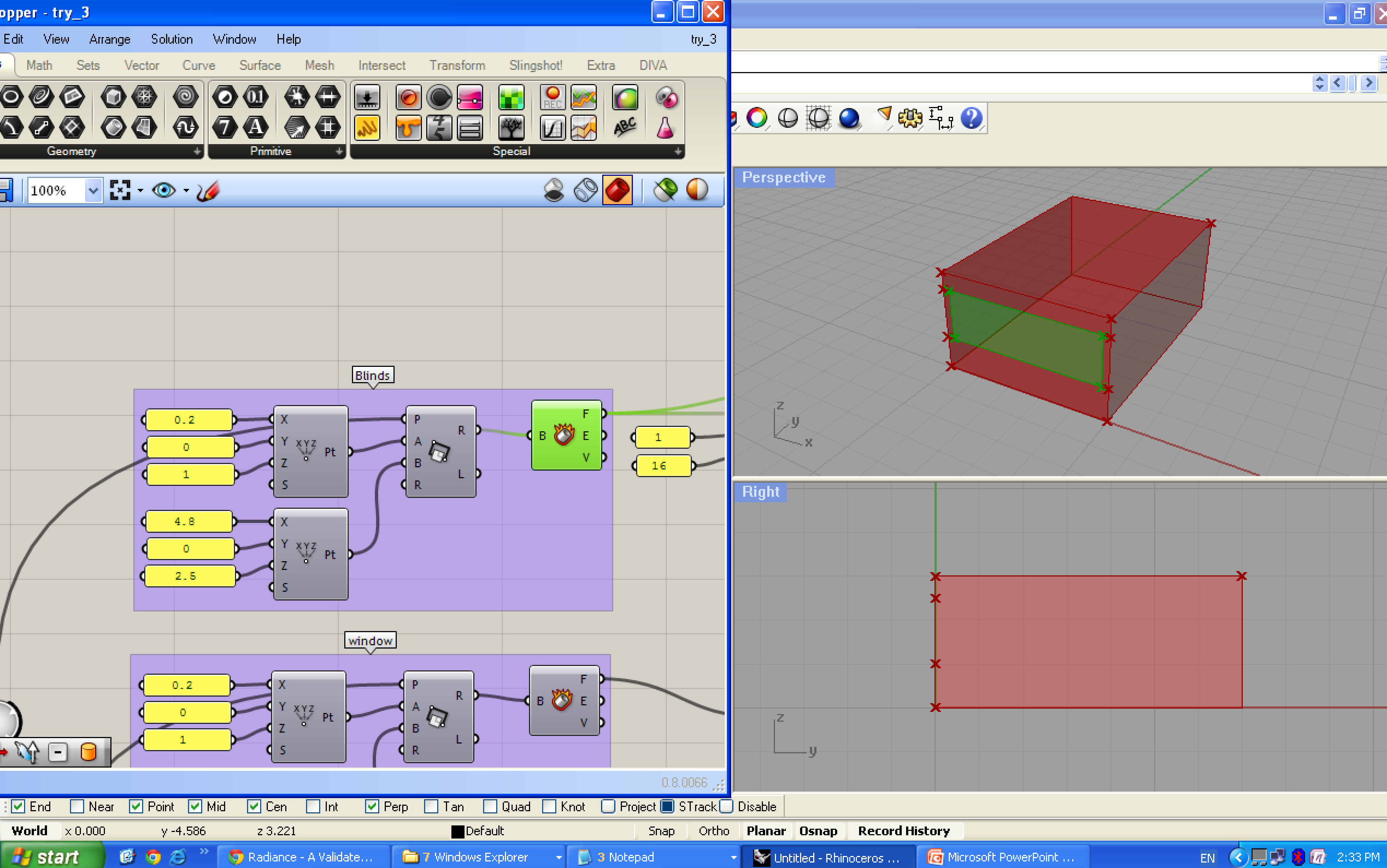
Exploding the geometry into surfaces to be assigned with the proper material



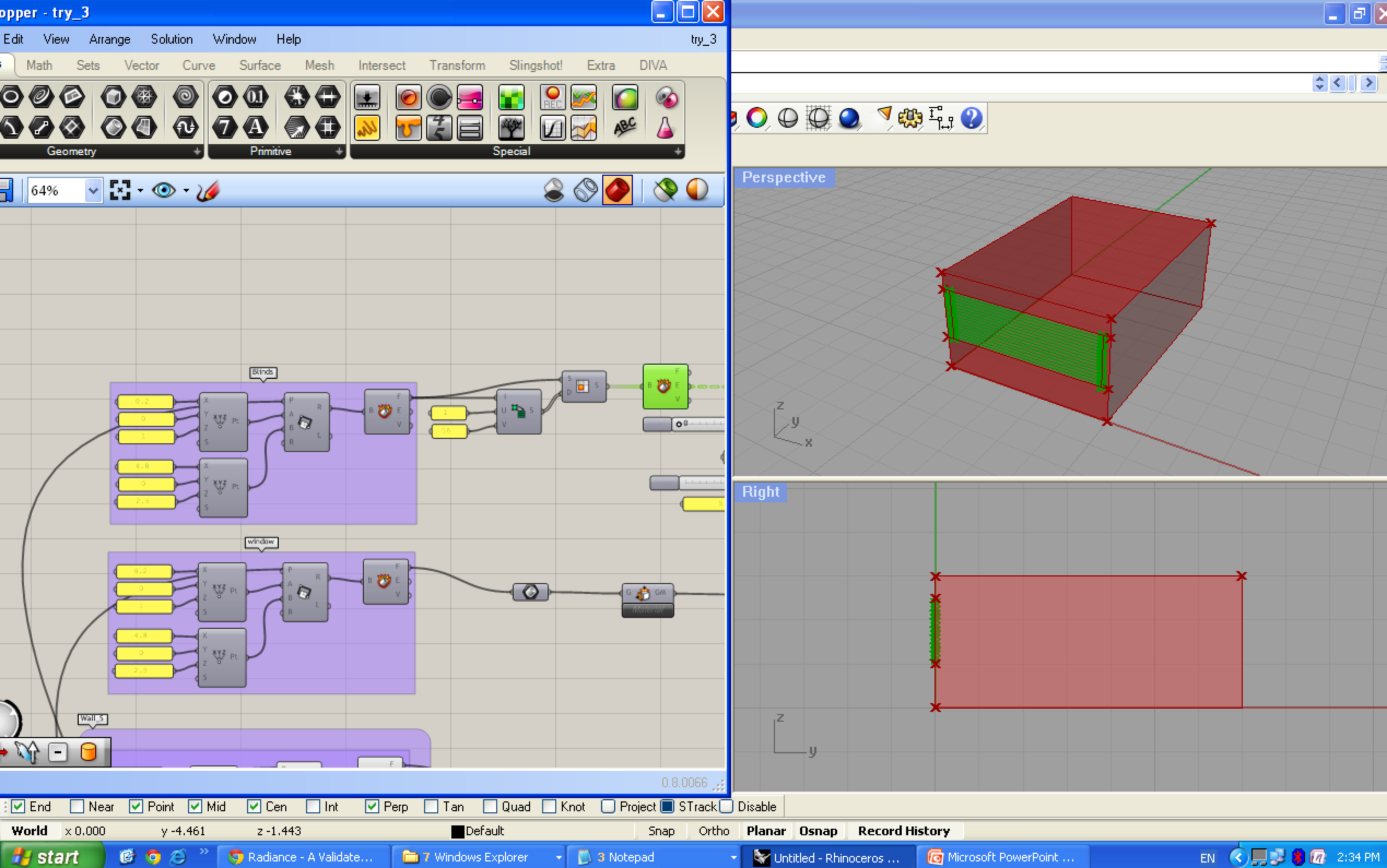
Meshing and assigning materials to different surfaces



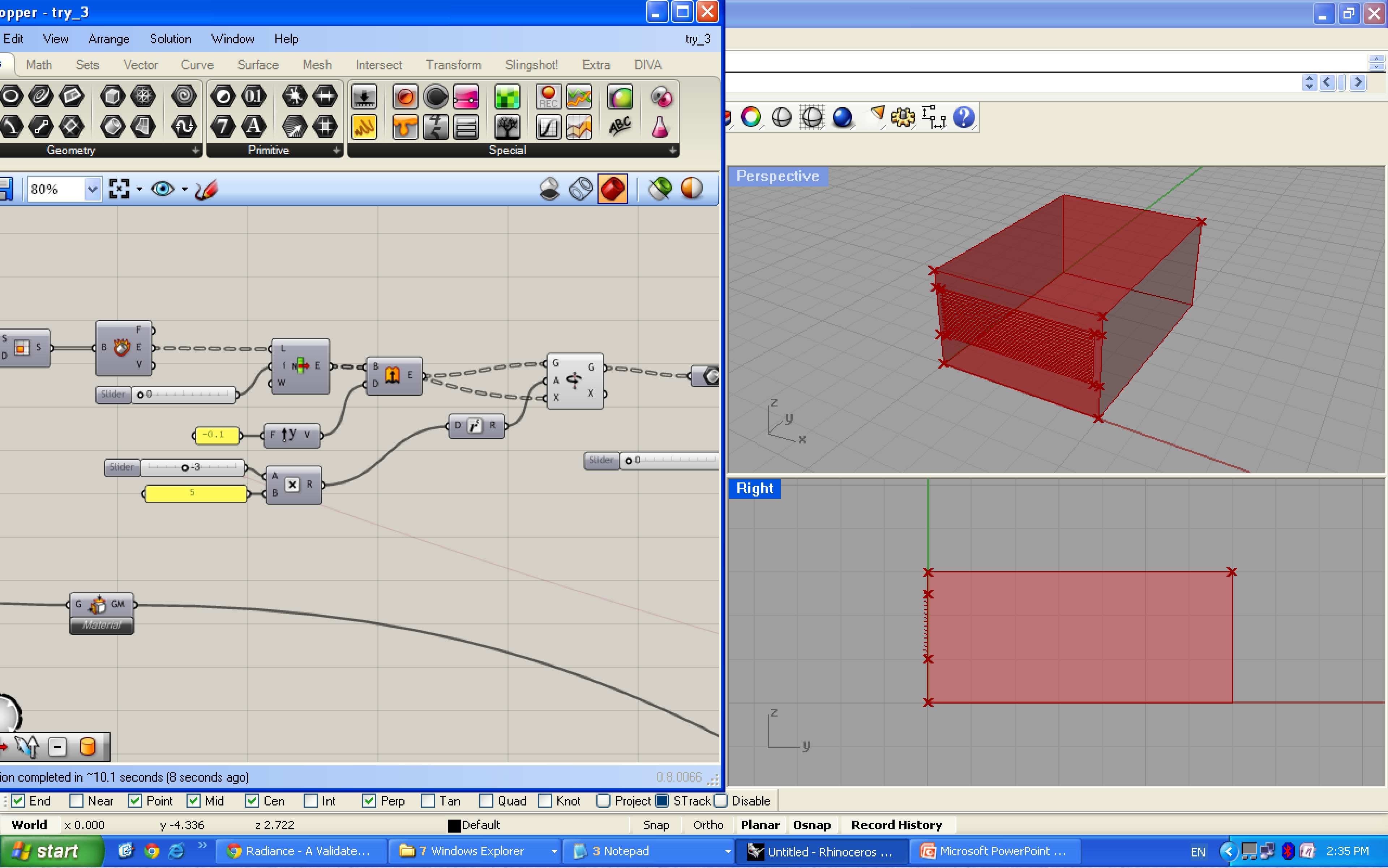
As for the south wall recreating the wall into 4 surfaces



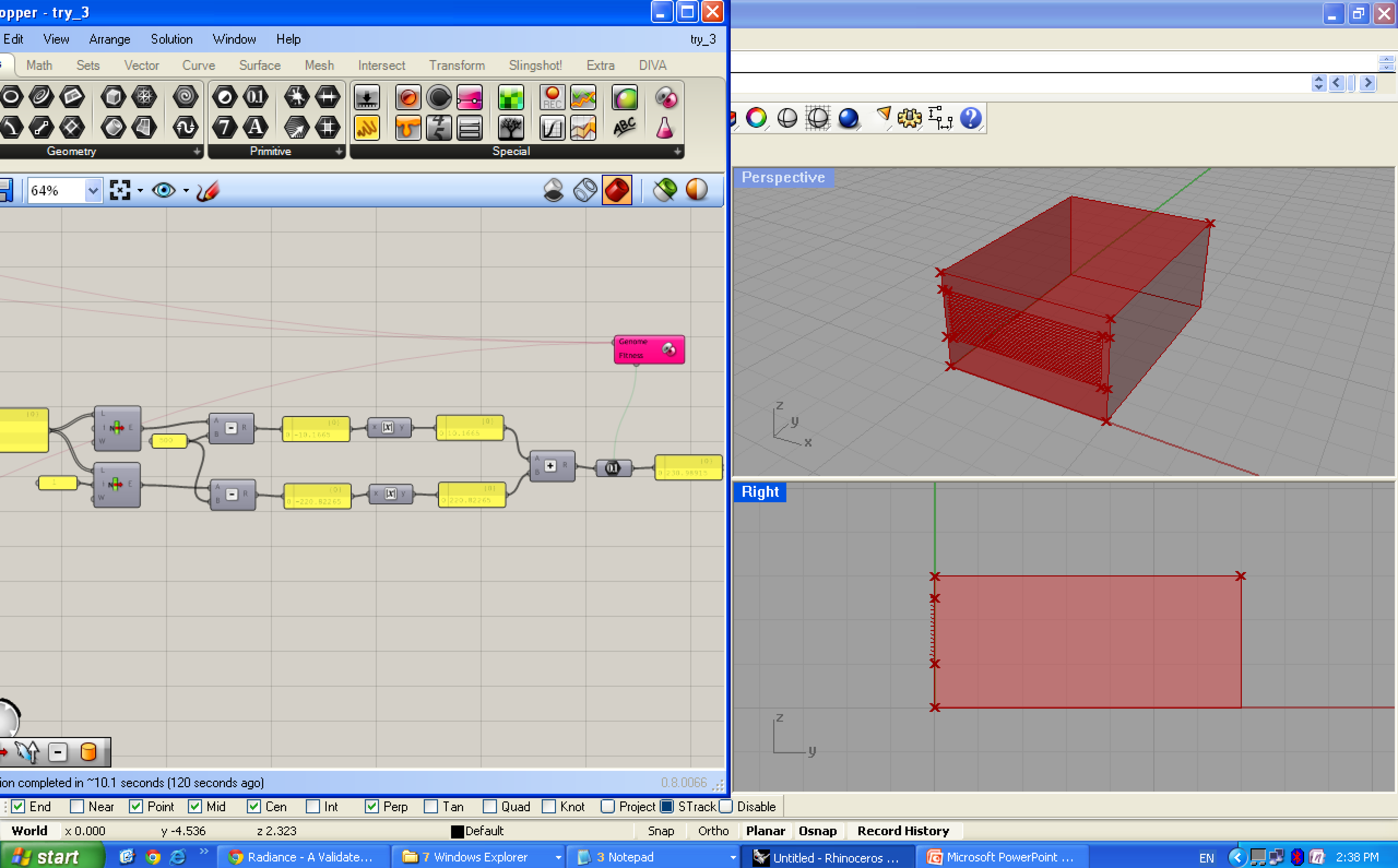
Creating the window plane



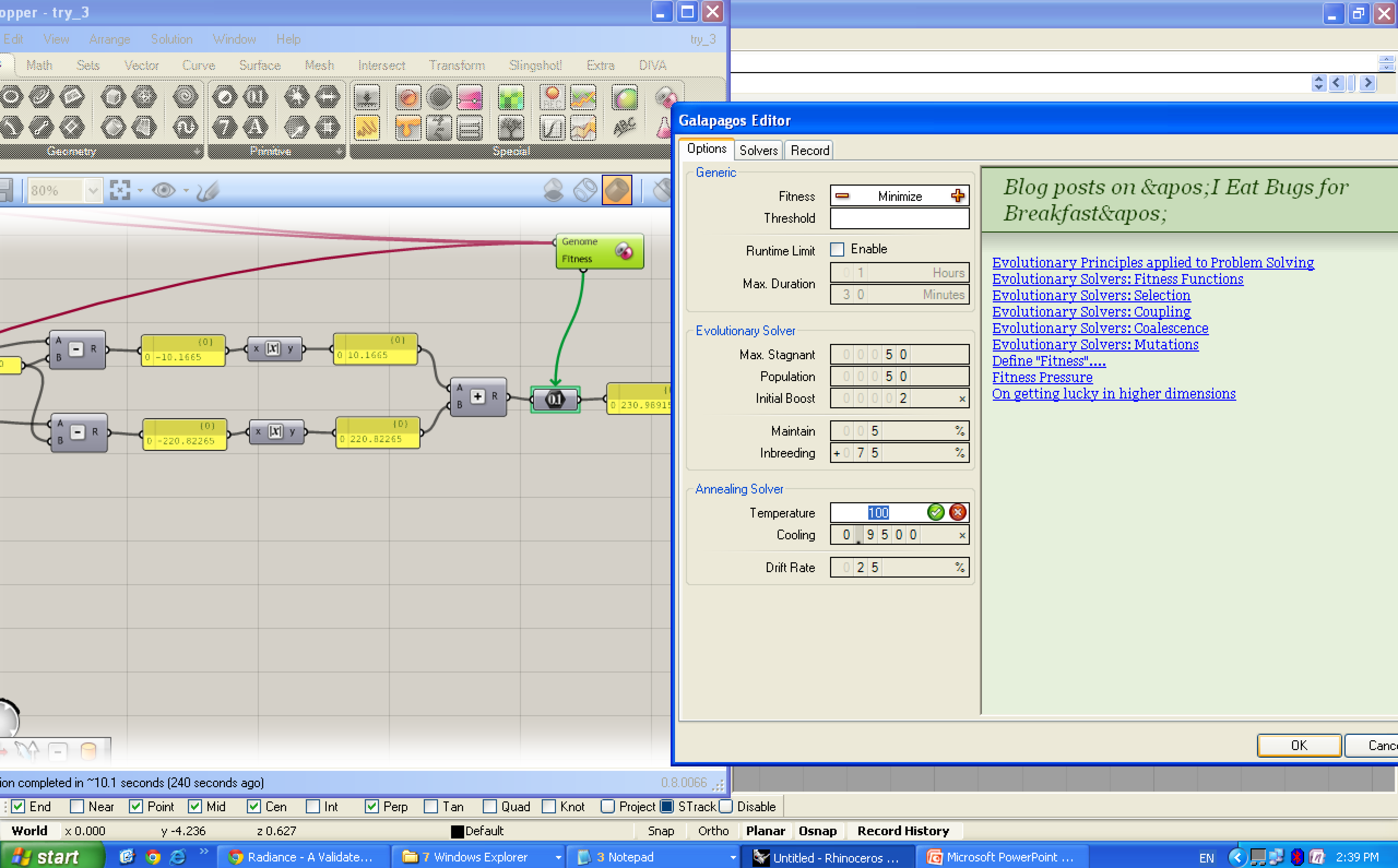
Recreating the blinds plane and subdividing into 16 horizontal surface



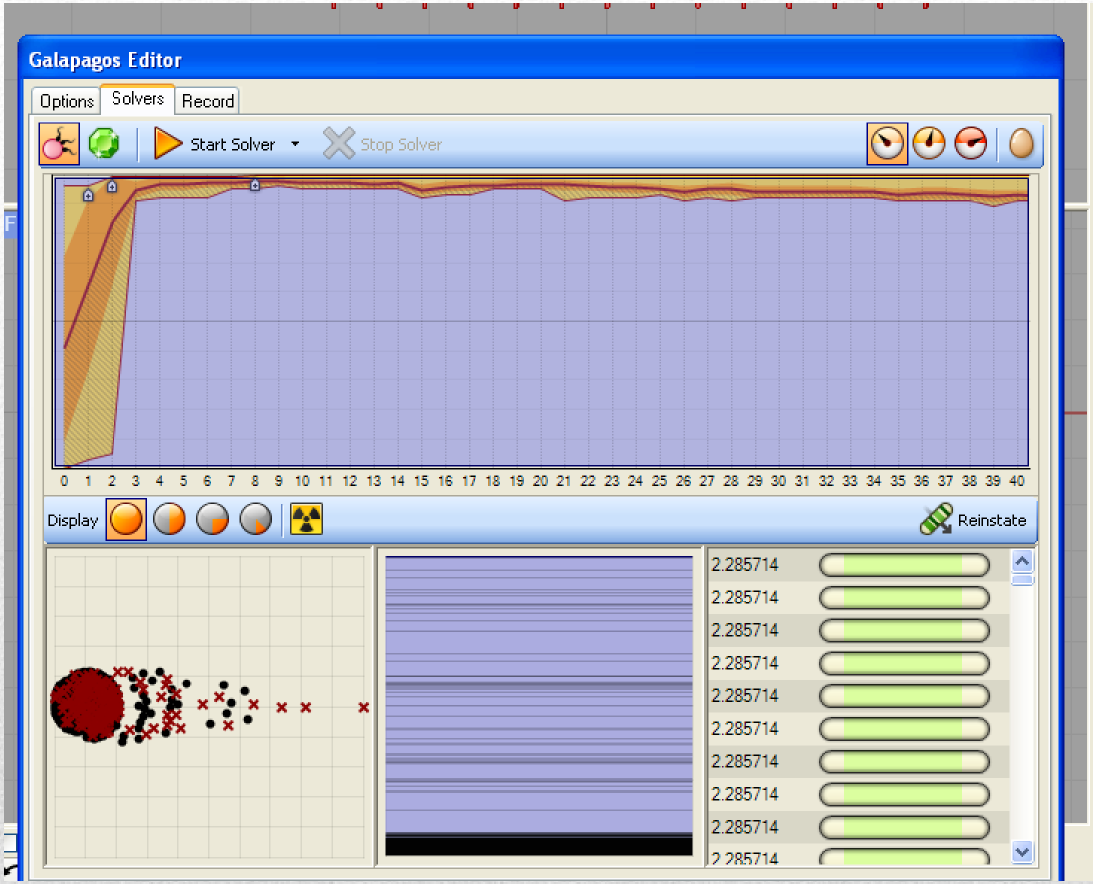
Adding the angle as a parameter defining the blinds



The fitness function



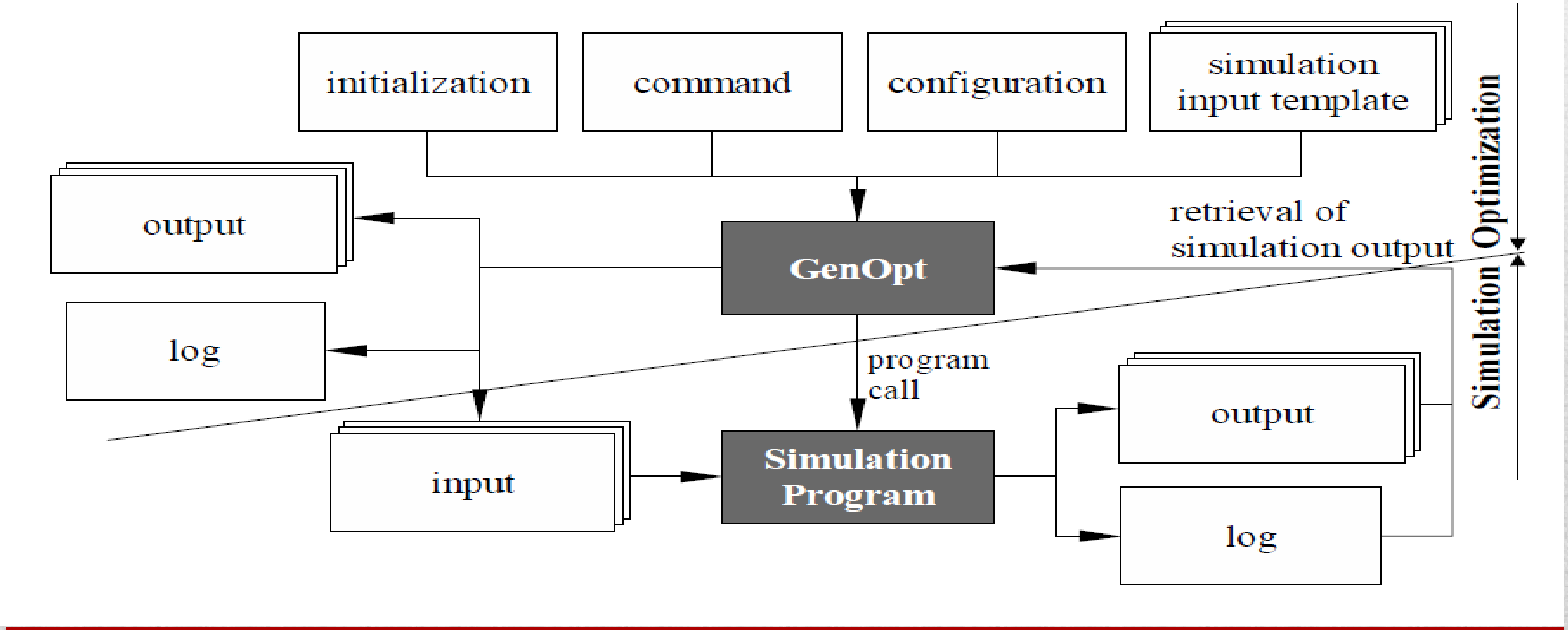
Start galapagos optimization


Galapagos Editor
Options
Solvers
Record
Start Solver
Stop Solver
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
Display
Reinstate
2.285714
2.285714
2.285714
2.285714
2.285714
2.285714
2.285714
2.285714
2.285714
2.285714

Results

ALTERNATIVE 2.

OPTIMIZATION USING GENOPT



- The purpose of GENOPT (GENeral OPTimization) is to enable an engineer to create a user-friendly system of programs for analyzing and/or optimizing anything

Interaction between GenOpt and Radiance

Radiance Files

1

Room_Template.rad

This is the typical .rad geometry file that is used in radiance. The only difference is that file replaces key values that will be used as design parameter with parameter names instead of the actual values

2

Clear_sky_Template.rad

This is the sky definition file that includes sky definition. This file may also contain simulation parameters if need be. So you may be able to rotate the sky

3

pp1_Template.pts

This is the file that contains the point locations to calculate. In our example only two points are used but many more points can be used

4

Run_Template.bat

This is the batch file that is called from within GenOpt with every simulation run

```
oconv clear_sky.rad room.rad>all.oct
rtrace –I+< pp1.pts  -ab 3 -ad 2000 -as 500 -ar 256 -aa 0.1 -ds .1 all.oct -opv octree |rcalc -0 ${$1} ${$2} ${$3} "result ${rec no} $[179*($1*0.265+$2*0.670+$3*0.065)]"> results.dat
:rview -vp 2.5 3 1 -vd -1 -1 0 -av .1 .1 .1 all.oct
```

luminance = 179 * (0.265*R + 0.670*G + 0.065*B)

The Radiance Batch File

1

```
oconv clear_sky.rad room.rad>all.oct
rtrace –I+< pp1.pts  -ab 3 -ad 2000 -as 500 -ar 256 -aa 0.1 -ds .1 all.oct -opv octree |rcalc -0 ${$1} ${$2} ${$3} "result ${rec no} $[179*($1*0.265+$2*0.670+$3*0.065)]"> results.dat
:rview -vp 2.5 3 1 -vd -1 -1 0 -av .1 .1 .1 all.oct
```

Load the Sky file and the model file

luminance = 179 * (0.265*R + 0.670*G + 0.065*B)

The Radiance Batch File

1

```
oconv clear_sky.rad room.rad>all.oct
rtrace –I+< pp1.pts  -ab 3 -ad 2000 -as 500 -ar 256 -aa 0.1 -ds .1 all.oct -opv
octree |rcalc -0 ${$1} ${$2} ${$3}
"result ${rec no}
$[179*($1*0.265+$2*0.670+$3*0.065)]"> results.dat
:rview -vp 2.5 3 1 -vd -1 -1 0 -av .1 .1 .1 all.oct
```

Ray trace the file using those parameters and calculate the results at the points in pp1.pts

luminance = 179 * (0.265*R + 0.670*G + 0.065*B)

The Radiance Batch File

1

```
oconv clear_sky.rad room.rad>all.oct
rtrace –I+< pp1.pts  -ab 3 -ad 2000 -as
500 -ar 256 -aa 0.1 -ds .1 all.oct -opv
octree |rcalc -0 ${$1} ${$2} ${$3}
"result ${rec no}
$[179*($1*0.265+$2*0.670+$3*0.06
5)]"> results.dat
:rview -vp 2.5 3 1 -vd -1 -1 0 -av .1 .1
.1 all.oct
```

Get the output values the R G B

luminance = 179 * (0.265*R + 0.670*G + 0.065*B)

The Radiance Batch File

1

The Radiance Batch File

1

```
oconv clear_sky.rad room.rad>all.oct
rtrace –I+< pp1.pts  -ab 3 -ad 2000 -as
500 -ar 256 -aa 0.1 -ds .1 all.oct -opv
octree |rcalc -0 ${$1} ${$2} ${$3}
"result ${rec no}
$[179*($1*0.265+$2*0.670+$3*0.06
5)]"> results.dat
:rview -vp 2.5 3 1 -vd -1 -1 0 -av .1 .1
.1 all.oct
```

Add “ result (number)” then use the R G B values in the equation to get the result in Lux

luminance = 179 * (0.265*R + 0.670*G + 0.065*B)

void plastic ceiling_mat
0
0
5 %refl_ceil% %refl_ceil% %refl_ceil% 0 0
void plastic floor_mat
0
05 0.3 0.3 0.3 0 0
void plastic south_wall_mat
0
0
5 0.6 0.6 0.6 0 0
..
..
..

Scen File Template 2

..
..
south_wall_mat polygon wall_S1
0
0
12
5 0 0
0 0 0
0 0 1
5 0 1
..
..
..

If we want to define geometric variables we want to define related variables so that is one changes the other will be calculated during runtime by Radiance, e.g.

Width of wall 1 = x1 + l

Then width of

Width of wall 2 = x2 + l

Scen File Template 2

```
#
# A set of grey venetian blinds
#
void plastic blind_grey
0
0
5          %refl_blinds%     %refl_blinds%     %refl_blinds%     0.02     0
!genblinds blind_grey blind 0.1 4.6 1.5 16 %angle_blinds% | xform -rz -90 -t 0.2 -0.1 1
..
..
..
```

Scen File Template ❷

```
!realc –n –e “val 1-%val 1% -o
objectif 1.func
```

Object1.functin

```
Mood  polygon wall_S1
0
0
12
5    ${val1 *2+1}    0
0    ${val1 *2+1}    0
0    ${val1 *2+1}    1
5    ${val1 *2+1}    1
..
```

Scen File Template 2

!realc –n –e “val 1-%val 1% -o objectif 1.func

Regenerate the model using the val1 as the coordinates (ex Y)

Object1.functin

Mood polygon wall_S1
0
0
12
5    ${val1 *2+1}   0
0    ${val1 *2+1}   0
0    ${val1 *2+1}   1
5    ${val1 *2+1}   1
..

Scen File Template   2

!realc –n –e “val 1-%val 1% -o objectif 1.func

Object1.functin

```
Mood  polygon wall_S1
0
0
12
5    ${val1 *2+1}    0
0    ${val1 *2+1}    0
0    ${val1 *2+1}    1
5    ${val1 *2+1}    1
..
```

Regenerate the model using the val1 as the coordinates of the room. From the object function

A java file can also be created to facilitates the results

http://vasilis.maheras.gr/?p=98&lang=en

Scen File Template 2

```
# gensky -ang 70 0 +s
# Ground ambient level: 20.4

void light solar
0
0
3 7.14e+006 7.14e+006 7.14e+006

solar source sun
0
0
4 0.000000 -0.342020 0.939693 0.5
…
```

Sky Definition Template

3

GenOpt Files

1

optWinXP.ini

This is the main file GenOpt and tells its where to find the actual project files, the results and the location of the objective function in the results. Also some post processing can be done here in the results

2

RadianceWinXP.cfg

This file is a small file that usually does not change with different projects. The only use of this file is to tell GenOpt how to call the program from the command line and the correct syntax for that.

3

Command.txt

This file has two main functions: 1. to tell GenOpt which parameters to change and how to change them (i.e. ranges, steps).

2. To specify the optimization algorithm to be used and its parameters

This file has following structure:

- Simulation
  - Files
    - Template
    - Input
    - Log
    - OutputConfiguration
  - CallParameter
  - Objective Function Location

- Optimization
  - Files
    - Command

Initialization file ❶

Simulation {

```
Files {
   Template {
File1 = room_Template.RAD;
File2 = clear_sky_Template.rad;
File3 = pp1_Template.pts;
File4 = run_Template.bat;
   }
```

Initialization File 1

```
Input {
    File1 = room.RAD;
SavePath1 = Save;
File2 = clear_sky.rad;
    SavePath2 = Save;
File3 = pp1.pts;
    SavePath3 = Save;
File4 = run.bat;
    SavePath4 = Save;
  }
```

Initialization File 1

```
Output {
    File1 = results.dat;
    SavePath1 = Save;
  }
  Configuration {
    File1 = "RadianceWinXP.cfg";
}
```

```
CallParameter { // optional section
  // The weather file without extension
  //Suffix = Cairo_IWEC;
 }
```

Initialization File 1

```
ObjectiveFunctionLocation
  {
Name1      = fitness;
Function1 = "add( %absPoi1% , %absPoi2% )";

Name2      = absPoi2;
Function2 = "abs( %Poi2-500% )";

Name3      = absPoi1;
Function3 = "abs( %Poi1-500% )";

Name4      = Poi2-500;
Function4 = "add( %Poi2% , -500 )";

Name5      = Poi1-500;
Function5 = "add( %Poi1% , -500 )";

Name6      = Poi2;
Delimiter6 = “001";

Name7      = Poi1;
Delimiter7 = “002";
  }
```

This section can be easily generated by a script file if you have many sensor points We developed a simple excel sheet that outputs this section automatically by clicking and dragging

The correct way of doing this is look at the delimiter that was inserted in the batch file

1

} // End Simulation section

Optimization {

```
Files {
    Command {
      File1 = command.txt;
    }
} // end of configuration file
```

} End Optimization

Initialization File 1

```
Vary{

Parameter{

Name    = refl_ceil;
Min     = 0;
Ini     = 0.5;
Max     = 1;
Step    = 0.2;
  }
..
..
```

This part defines the parameters and how they vary

Command File

2

```
OptimizationSettings{
 MaxIte = 2500;
 MaxEqualResults = 1000;
 WriteStepNumber = false;
 UnitsOfExecution = 0;
}


Algorithm{ Main =
GeneticAlgorithmJGAP;
MaxGenerations = 20; // 0 <
maxGenerations

..

..
```

This part defines the parameters of the simulation

Command File

2

SimulationStart {

Command = "cmd /x /c run.bat
%Simulation.Files.Input.File1%
%Simulation.CallParameter.Suffix%\"";

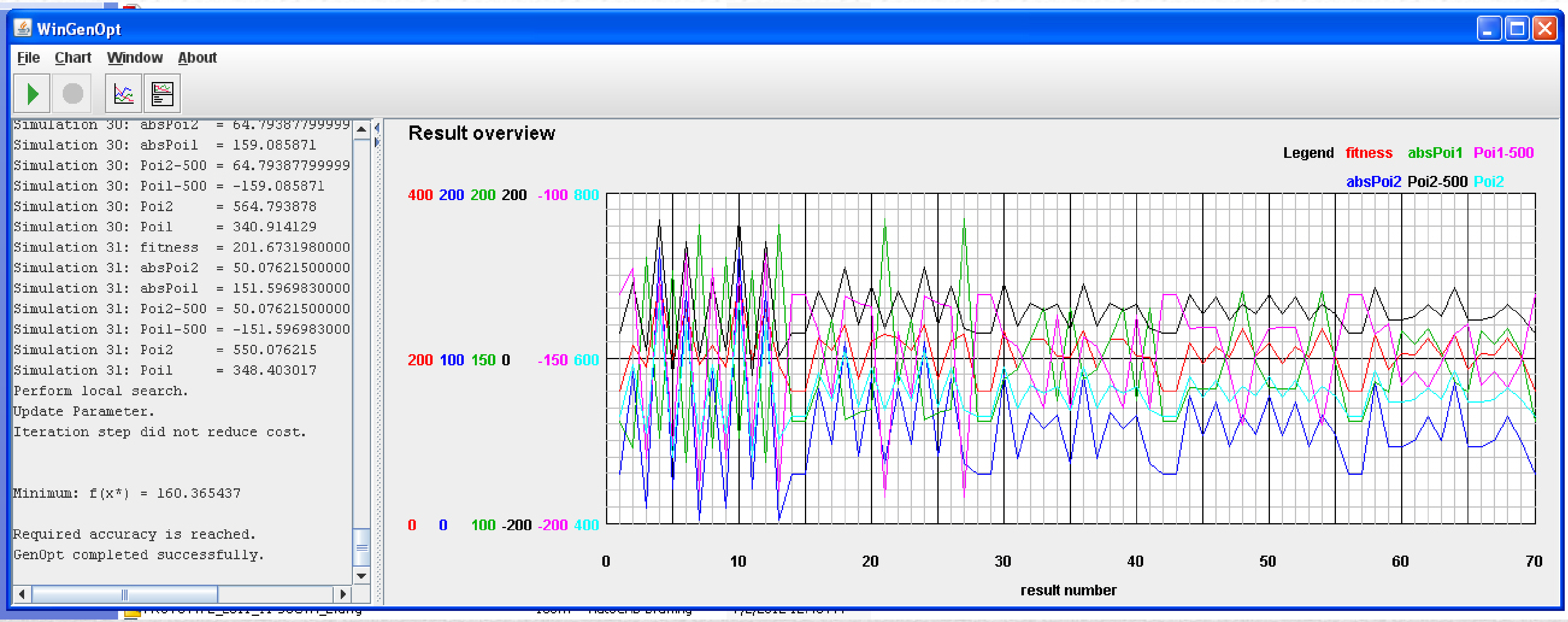
..

..

}

If you wanted access to any of the files you defined in the initialization file you can access them here

Radiance win Config 3


WinGenOpt
File Chart Window About
Simulation 30: absPoi2 = 64.79387799999
Simulation 30: absPoi1 = 159.085871
Simulation 30: Poi2-500 = 64.79387799999
Simulation 30: Poi1-500 = -159.085871
Simulation 30: Poi2 = 564.793878
Simulation 30: Poi1 = 340.914129
Simulation 31: fitness = 201.6731980000
Simulation 31: absPoi2 = 50.07621500000
Simulation 31: absPoi1 = 151.5969830000
Simulation 31: Poi2-500 = 50.07621500000
Simulation 31: Poi1-500 = -151.596983000
Simulation 31: Poi2 = 550.076215
Simulation 31: Poi1 = 348.403017
Perform local search.
Update Parameter.
Iteration step did not reduce cost.
Minimum: f(x*) = 160.365437
Required accuracy is reached.
GenOpt completed successfully.
Result overview
Legend fitness absPoi1 Poi1-500
absPoi2 Poi2-500 Poi2
400 200 200 200 -100 800
200 100 150 0 -150 600
0 0 100 -200 -200 400
0 10 20 30 40 50 60 70
result number

Results

Thank You