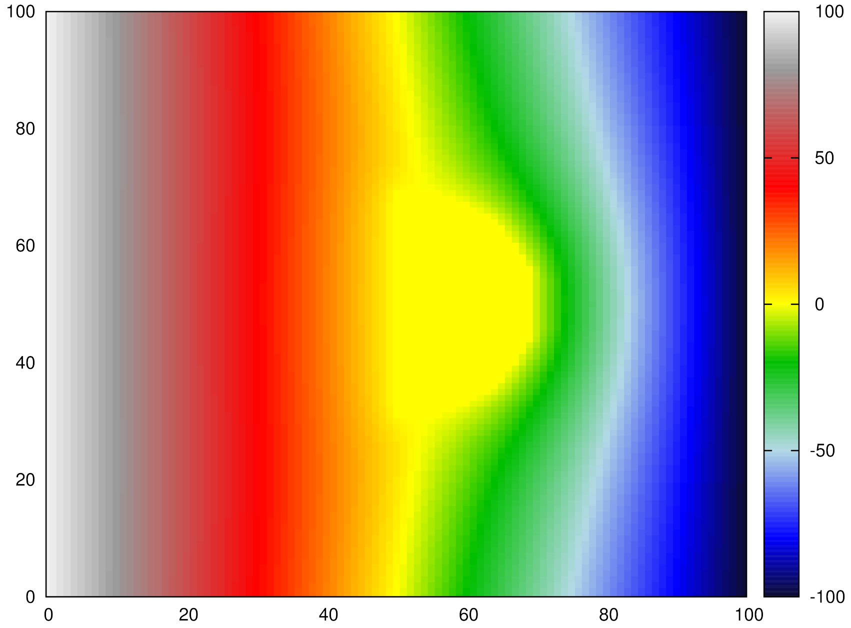
Using the solver:

```
#include ”finite_difference.h”

using namespace std;

int main() {

        int n = 100;
        int m = 100;

        Grid first_grid;
        first_grid.load_grid(n,m);
        first_grid.set_flow(100, −100);
        first_grid.set_halfcircle_east(50,50,20,0);

        Finite_Difference fd;
        fd.to_solve(first_grid);
        fd.set_precision(0);
        fd.solve();
        Grid sol = fd.get_solution();
        //cout << fd.number_of_iterations() << endl;
        sol.gnuplot_values();


        return 0;

        }
```

Creating your own boundary functions:

```cpp
void set_circle(int x, int y, unsigned int r, double val) {
        if(x - r < 0 || x + r > values.size() - 1 || y - r < 0 ||
        y + r > values[0].size() - 1)cout << "Out of range." << endl;
        else {
                for (int xs = x-r; xs<=x+r; xs++) {
                        for(int ys = y-r; ys<=y+r; ys++) {
                        Coordinate xy;
                        xy.set_xy(xs,ys);
                        Coordinate mid;
                        mid.set_xy(x,y);
                        if( mid.distance(xy) < r )
                        {values[xs][ys].value = val;
                        values[xs][ys].boundary = true;}
                        }
                }
        }
}
```

Plotting in gnuplot:



Use *grid.gnuplot_values*() and don't print anything else in your program.

Redirect output to data file (*./my_program* > *data.dat*)

gnuplot > plot 'data.dat' matrix with image

Various options possible, example file on github.