

Final Assignment: MATLAB and C++

Jordan Lian

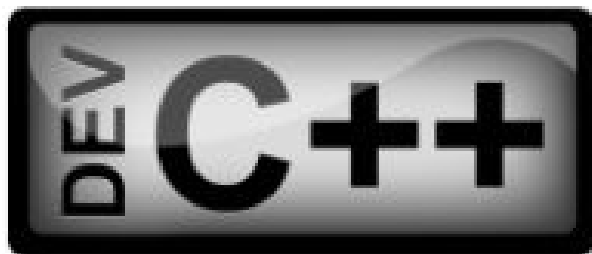
GE 1502 -- 308 Hurtig Hall

10:30 - 11:35am

Professor Whalen

Table of Contents

1. Letter
2. Benchmark Plot
3. Weld Radius vs Time to Steady State
 - a. Tin / Lead (60/40)
 - b. Tin Lead (50/50)



Letter: Business-Style Letter of Transmittal

Jordan Lian

10 Forsyth Street, #1190
Boston, MA 02115
(610) 425-8865
lian.jo@husky.neu.edu

March 26, 2018

Professor Richard Whalen
Northeastern University, MIE Department
Boston, MA 02115

Dear Professor Whalen,

Hope you are doing well. In this final assignment combining MATLAB and C++, I learned to use C++ to create a data table using while loops and some embedded logic and then MATLAB to input the table and plot out a graph based on the values using array, fscan, max, and hold on/off. There were a couple constraints I had to work with, such as a maximum time of 5 seconds, and a maximum difference of 0.1 for liquid and current temperature.

For the benchmark plot, I wrote two identical programs that outputted the results to separate txt files since there were two materials tested (Tin / Lead 60-40 and 50-50). The resulting plots were not linear, but instead they were inverse functions following the pattern of $f(x) = 1/x$. This graph is an inverse graph because as time goes by, the more heat is transferred from the material, and the temperature is reduced as a result. So as time increases, temperature reduces. This will always result in an inverse graph.

For the weld radius versus time to steady state, I also created identical code for different txt files for each material. Plotting the data in MATLAB was identical to the plotting for the benchmark plot with only different txt files. The resulting plots were linear, with similar slopes. However, the 50-50 tin/lead had a much higher maximum radius of 0.00137 m (5.015 s) whereas the 60-40 tin/lead had a maximum radius of 0.00089 m (5.026 s). The 50-50 tin/lead also has the fastest response time of 1.829 s (0.0005 m radius) where the 60-40 lead's fastest response time was 2.822 s (0.0005 m radius). If cost wasn't a factor and the 50-50 tin/lead material was the

stronger material of the two, then the 50-50 tin/lead should be used rather than the 60-40 tin/lead due to the faster response time and a larger maximum radius.

My results are included below this letter. I look forward to seeing your evaluation of my assignment.

Best regards,

Jordan Lian

Benchmark Plot

Tin / Lead 60/40

C++ source code

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>

using namespace std;

//declare subfunction
void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC);

int main(void)
{
    double pi;
    double TI, TL, h, density, c, r , time_step, change_in_temperature;
    double As, volume, m, T, ROC;
    double time = 0;
    int i;

    ofstream outfile;
    outfile.open("benchmark_plot_60-40.txt");

    pi = 4*atan(1);

    cout << "Enter initial temperature of thermocouple junction (sphere) (C) --> ";
```

```

cin >> TI;

cout << "Enter liquid temperature (C) --> ";
cin >> TL;

cout << "Enter heat transfer coefficient (W/m^2*C) --> ";
cin >> h;

cout << "Enter sphere density (kg/m^3) --> ";
cin >> density;

cout << "Enter sphere specific heat (J / kg * C)--> ";
cin >> c;

cout << "Enter sphere radius (m) --> ";
cin >> r;

cout << "Enter desired time step for temperature history (s) --> ";
cin >> time_step;

//for loop -- time increment

T = TI;

cout << "Time (s) \tTemp (C)\n";

while( (abs(TL - T) >= 0.1) && (time <= 5) )
{
    cout << time << " \t\t " << T << endl;
    outfile << time << " \t\t " << T << endl;

    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;
}

cout << endl;

```

```

    system("pause");
    return 0;
}

```

//sub-function -- rate of change

```

void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC)
{
    double pi = 4 * atan(1);
    double As = 4*pi*r*r; //surface area
    double volume = pi*r*r*r*(4.0/3.0); //volume
    double m = density * volume;

    ROC = (h*As*(TL - T) ) / (m*c);
}

```

C++ results

```

C:\Northwestern University\Dev C++\GE 1502\Benchmark Plot 60-40.exe
Enter initial temperature of thermocouple junction (sphere) (C) --> 100
Enter liquid temperature (C) --> 25
Enter heat transfer coefficient (W/m^2C) --> 1000
Enter sphere density (kg/m^3) --> 8922
Enter sphere specific heat (J / kg * C)--> 287
Enter sphere radius (m) --> 0.0005
Enter desired time step for temperature history (s) --> 0.001
Time (s)      Temp (C)
0             100
0.001         99.8243
0.002         99.6489
0.003         99.474
0.004         99.2995
0.005         99.1254
0.006         98.9517
0.007         98.7784
0.008         98.6056
0.009         98.4331
0.01          98.261
0.011         98.0894
0.012         97.9181
0.013         97.7472
0.014         97.5768
0.015         97.4067
0.016         97.2371
0.017         97.0678
0.018         96.8989
0.019         96.7304
0.02          96.5624
0.021         96.3947
0.022         96.2274
0.023         96.0605
0.024         95.894
0.025         95.7279
0.026         95.5621
0.027         95.3968
0.028         95.2318
0.029         95.0673
0.03          94.9031
0.031         94.7393
0.032         94.5759
0.033         94.4129
0.034         94.2502
0.035         94.0879
0.036         93.9261
0.037         93.7646
0.038         93.6034
0.039         93.4427
0.04          93.2823
0.041         93.1223

```

```
C:\Northeastern University\Dev C++\GE 1502\Benchmark Plot 60-40.exe
2.781      25.1101
2.782      25.1098
2.783      25.1096
2.784      25.1093
2.785      25.1091
2.786      25.1088
2.787      25.1085
2.788      25.1083
2.789      25.108
2.79       25.1078
2.791      25.1075
2.792      25.1073
2.793      25.107
2.794      25.1068
2.795      25.1065
2.796      25.1063
2.797      25.106
2.798      25.1058
2.799      25.1055
2.8        25.1053
2.801      25.105
2.802      25.1048
2.803      25.1045
2.804      25.1043
2.805      25.1041
2.806      25.1038
2.807      25.1036
2.808      25.1033
2.809      25.1031
2.81       25.1028
2.811      25.1026
2.812      25.1024
2.813      25.1021
2.814      25.1019
2.815      25.1016
2.816      25.1014
2.817      25.1012
2.818      25.1009
2.819      25.1007
2.82       25.1005
2.821      25.1002
Press any key to continue . . .
```

Tin / Lead 50/50

C++ source code

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>
```

```
using namespace std;
```

```
//declare subfunction
```

```
void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC);
```

```
int main(void)
```

```
{
```

```
    double pi;
```

```
    double TI, TL, h, density, c, r , time_step, change_in_temperature;
```

```
    double As, volume, m, T, ROC;
```

```
    double time = 0;
```

```
    int i;
```

```
    ofstream outfile;
```

```
    outfile.open("benchmark_plot_50-50.txt");
```

```

pi = 4*atan(1);

cout << "Enter initial temperature of thermocouple junction (sphere) (C) --> ";
cin >> TI;

cout << "Enter liquid temperature (C) --> ";
cin >> TL;

cout << "Enter heat transfer coefficient (W/m^2*C) --> ";
cin >> h;

cout << "Enter sphere density (kg/m^3) --> ";
cin >> density;

cout << "Enter sphere specific heat (J / kg * C)--> ";
cin >> c;

cout << "Enter sphere radius (m) --> ";
cin >> r;

cout << "Enter desired time step for temperature history (s) --> ";
cin >> time_step;

//for loop -- time increment

T = TI;

cout << "Time (s) \tTemp (C)\n";

while( (abs(TL - T) >= 0.1) && (time <= 5) )
{
    cout << time << " \t\t " << T << endl;
    outfile << time << " \t\t " << T << endl;

    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;
}

```

```

    }

    cout << endl;

    system("pause");
    return 0;
}

//sub-function -- rate of change

```

```

void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC)
{
    double pi = 4 * atan(1);
    double As = 4*pi*r*r; //surface area
    double volume = pi*r*r*r*(4.0/3.0); //volume
    double m = density * volume;

    ROC = (h*As*(TL - T) ) / (m*c);
}

```

C++ results

The screenshot shows a Windows command prompt window titled "C:\Northwestern University\Dev C++\GE 1502\Benchmark Plot 50-50.exe". The program prompts for several input values: initial temperature of thermocouple junction (sphere) (C) --> 100, liquid temperature (C) --> 25, heat transfer coefficient (W/m^2°C) --> 1000, sphere density (kg/m^3) --> 9325, sphere specific heat (J / kg * C) --> 178, sphere radius (m) --> 0.0005, and desired time step for temperature history (s) --> 0.001. Below the prompts, a table displays the temperature history over time.

Time (s)	Temp (C)
0	100
0.001	99.7289
0.002	99.4588
0.003	99.1896
0.004	98.9214
0.005	98.6542
0.006	98.388
0.007	98.1227
0.008	97.8584
0.009	97.595
0.01	97.3326
0.011	97.0711
0.012	96.8106
0.013	96.551
0.014	96.2924
0.015	96.0347
0.016	95.7779
0.017	95.5221
0.018	95.2671
0.019	95.0131
0.02	94.76
0.021	94.5079
0.022	94.2566
0.023	94.0063
0.024	93.7568
0.025	93.5083
0.026	93.2606
0.027	93.0139
0.028	92.768
0.029	92.5231
0.03	92.279
0.031	92.0358
0.032	91.7935
0.033	91.552
0.034	91.3115
0.035	91.0718
0.036	90.8329
0.037	90.595
0.038	90.3578
0.039	90.1216
0.04	89.8862
0.041	89.6516


```
C:\Northeastern University\Dev C++\GE 1502\Benchmark Plot 50-50.exe
1.784      25.1173
1.785      25.1169
1.786      25.1165
1.787      25.116
1.788      25.1156
1.789      25.1152
1.79       25.1148
1.791      25.1144
1.792      25.114
1.793      25.1135
1.794      25.1131
1.795      25.1127
1.796      25.1123
1.797      25.1119
1.798      25.1115
1.799      25.1111
1.8        25.1107
1.801      25.1103
1.802      25.1099
1.803      25.1095
1.804      25.1091
1.805      25.1087
1.806      25.1083
1.807      25.1079
1.808      25.1075
1.809      25.1072
1.81       25.1068
1.811      25.1064
1.812      25.106
1.813      25.1056
1.814      25.1052
1.815      25.1048
1.816      25.1045
1.817      25.1041
1.818      25.1037
1.819      25.1033
1.82       25.103
1.821      25.1026
1.822      25.1022
1.823      25.1019
1.824      25.1015
1.825      25.1011
1.826      25.1008
1.827      25.1004
1.828      25.1
Press any key to continue . . .
```

MATLAB m-file

```
fid = fopen('benchmark_plot_60-40.txt', 'r');
```

```
formatSpec = '%f';
```

```
s = [2 Inf];
```

```
array = fscanf(fid, formatSpec, s);
```

```
length = max(size(array));
```

```
for i = 1 : length
```

```
    time1(i) = array(1, i);
```

```
    temp1(i) = array(2, i);
```

```
end
```

```
plot(time1, temp1);
```

```
xlabel('time (s)');
```

```
ylabel('temperature (C)');
```

```
title('Benchmark Plot');
```

```
hold on
```

```
fid = fopen('benchmark_plot_50-50.txt', 'r');
```

```
formatSpec = '%f';
```

```

s = [2 Inf];

array = fscanf(fid, formatSpec, s);
length = max(size(array));

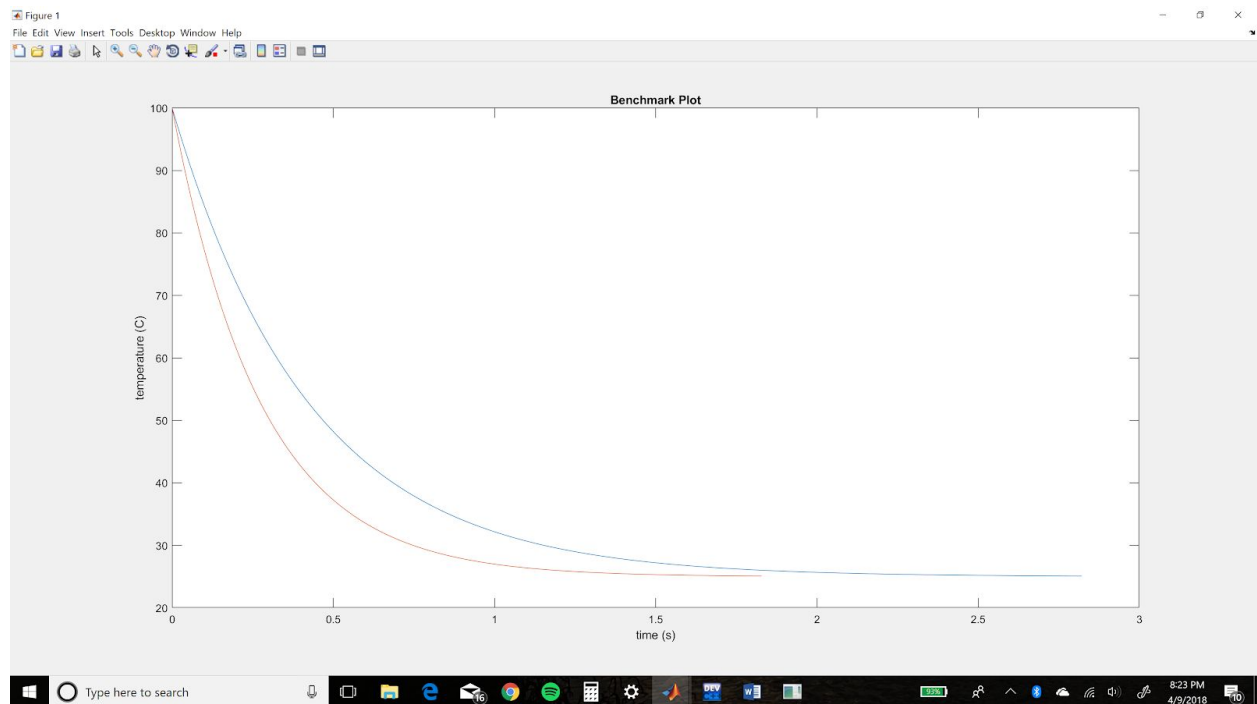
for i = 1 : length
    time2(i) = array(1, i);
    temp2(i) = array(2, i);
end

plot(time2, temp2);
hold off

```

Combined MATLAB plot

Note: **Red** = Tin / Lead 50-50, **Blue** = Tin / Lead 60-40



Weld Radius vs Time to Steady State

Tin / Lead 60/40

C++ source code

```

#include <iostream>
#include <cstdlib>
#include <cmath>

```

```

#include <fstream>

using namespace std;

//declare subfunction
void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC);

int main(void)
{
    //declare variables
    double pi;
    double TI, TL, h, density, c, time_step, change_in_temperature;
    double As, volume, m, T, ROC, r, r_increment;
    double time = 0;
    int i, d = 1;

    //output file for results
    ofstream outfile;
    outfile.open("weld_radius_time_60-40.txt"); //60-40 txt file

    pi = 4*atan(1); //define pi

    //get values from user
    cout << "Enter initial temperature of thermocouple junction (sphere) (C) --> ";
    cin >> TI;

    cout << "Enter liquid temperature (C) --> ";
    cin >> TL;

    cout << "Enter heat transfer coefficient (W/m^2*C) --> ";
    cin >> h;

    cout << "Enter sphere density (kg/m^3) --> ";
    cin >> density;

    cout << "Enter sphere specific heat (J / kg * C) --> ";
    cin >> c;

    cout << "Enter sphere radius (m) --> ";
    cin >> r;

    cout << "Enter desired time step for temperature history (s) --> ";

```

```

cin >> time_step;

T = TI;
r_increment = 0.00001;
cout << "\nTime (s) \tWeld Radius(m)\n";
time = 0;

while(time <= 5) //overall constraint
{
    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;

    //once temperature reaches limit, record time and corresponding radius
    if(abs(TL - T) <= 0.1)
    {
        //print out results
        cout << time << " \t\t " << r << endl;
        outfile << time << " \t\t " << r << endl;

        //reset/increment variables
        r = r + r_increment; //increment radius for new cycle
        time = 0; //reset time to 0 for new cycle
        T = TI; //reset temperature
    }
}

do //execute loop one last time
{
    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;

    //logic test to break from loop and finish program
    if(abs(TL - T) <= 0.1)

```

```

        {
            //print out results
            cout << time << " \t\t " << r << endl;
            outfile << time << " \t\t " << r << endl;

            //break from loop
            d = 2;
        }
    } while(d == 1);

    cout << endl;

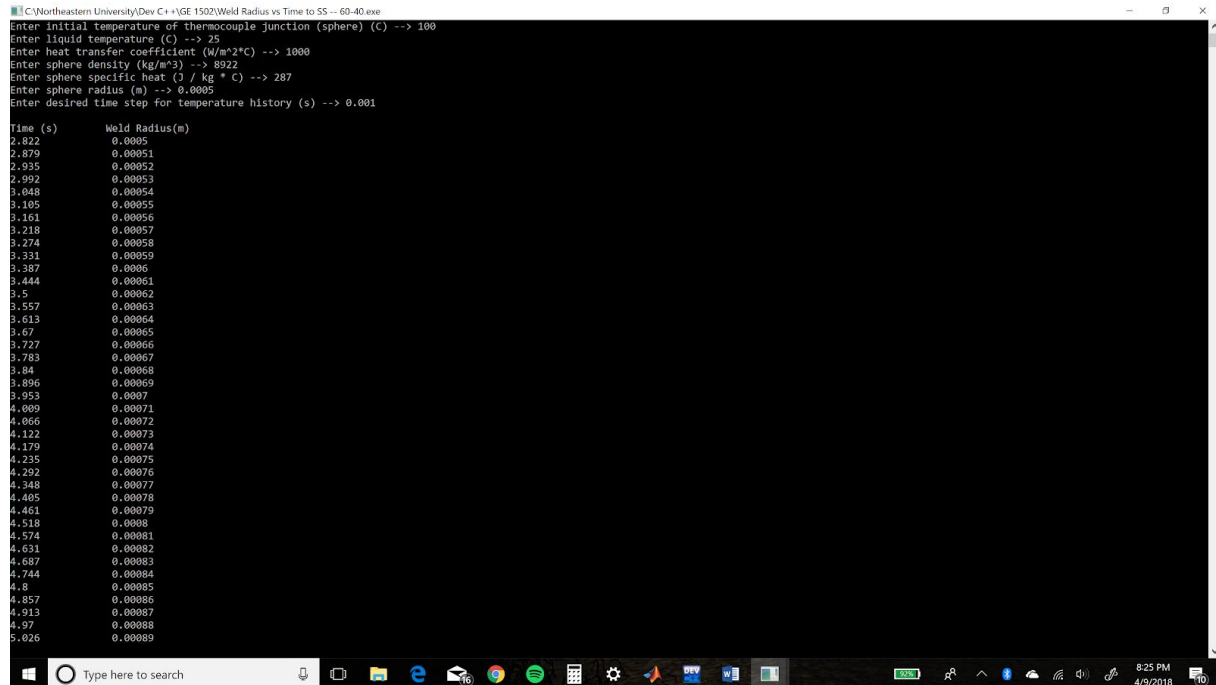
    system("pause");
    return 0;
}

//sub-function -- rate of change
void rate_of_change(double h, double r, double density, double TL, double T, double c, double
&ROC)
{
    double pi = 4 * atan(1);
    double As = 4*pi*r*r; //surface area
    double volume = pi*r*r*r*(4.0/3.0); //volume
    double m = density * volume;

    ROC = (h*As*(TL - T) ) / (m*c);
}

```

C++ results



The screenshot shows a Windows command prompt window titled "C:\Northwestern University\Dev C++\GE 1503\Weld Radius vs Time to SS -- 69-40.exe". The program prompts for several input values: initial temperature of thermocouple junction (sphere) (C) --> 100, liquid temperature (C) --> 25, heat transfer coefficient (W/m^2*C) --> 1000, sphere density (kg/m^3) --> 8922, sphere specific heat (J / kg * C) --> 287, sphere radius (m) --> 0.0005, and desired time step for temperature history (s) --> 0.001. Below the prompts, a table is displayed with two columns: "Time (s)" and "Weld Radius(m)". The table contains 30 rows of data, showing a steady increase in both time and weld radius from 0.822 to 5.026 seconds and 0.0005 to 0.00089 meters respectively.

Time (s)	Weld Radius(m)
0.822	0.0005
0.879	0.00051
0.935	0.00052
0.992	0.00053
1.048	0.00054
1.105	0.00055
1.161	0.00056
1.218	0.00057
1.274	0.00058
1.331	0.00059
1.387	0.0006
1.444	0.00061
1.5	0.00062
1.557	0.00063
1.613	0.00064
1.67	0.00065
1.727	0.00066
1.783	0.00067
1.84	0.00068
1.896	0.00069
1.953	0.0007
2.009	0.00071
2.066	0.00072
2.122	0.00073
2.179	0.00074
2.235	0.00075
2.292	0.00076
2.348	0.00077
2.405	0.00078
2.461	0.00079
2.518	0.0008
2.574	0.00081
2.631	0.00082
2.687	0.00083
2.744	0.00084
2.8	0.00085
2.857	0.00086
2.913	0.00087
2.97	0.00088
3.026	0.00089

Tin / Lead 50/50

C++ source code

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>
```

```
using namespace std;
```

```
//declare subfunction
```

```
void rate_of_change(double h, double r, double density, double TL, double T, double c, double &ROC);
```

```
int main(void)
```

```
{
```

```
    //declare variables
```

```
    double pi;
```

```
    double TI, TL, h, density, c, time_step, change_in_temperature;
```

```
    double As, volume, m, T, ROC, r, r_increment;
```

```
    double time = 0;
```

```
    int i, d = 1;
```

```

//output file for results
ofstream outfile;
outfile.open("weld_radius_time_50-50.txt"); //60-40 txt file

pi = 4*atan(1); //define pi

//get values from user
cout << "Enter initial temperature of thermocouple junction (sphere) (C) --> ";
cin >> TI;

cout << "Enter liquid temperature (C) --> ";
cin >> TL;

cout << "Enter heat transfer coefficient (W/m^2*C) --> ";
cin >> h;

cout << "Enter sphere density (kg/m^3) --> ";
cin >> density;

cout << "Enter sphere specific heat (J / kg * C) --> ";
cin >> c;

cout << "Enter sphere radius (m) --> ";
cin >> r;

cout << "Enter desired time step for temperature history (s) --> ";
cin >> time_step;

T = TI;
r_increment = 0.00001;
cout << "\nTime (s) \tWeld Radius(m)\n";
time = 0;

while(time <= 5) //overall constraint
{
    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;
}

```

```

//once temperature reaches limit, record time and corresponding radius
if(abs(TL - T) <= 0.1)
{
    //print out results
    cout << time << " \t\t " << r << endl;
    outfile << time << " \t\t " << r << endl;

    //reset/increment variables
    r = r + r_increment; //increment radius for new cycle
    time = 0; //reset time to 0 for new cycle
    T = TL; //reset temperature
}
}

do //execute loop one last time
{
    //call sub-function, 7 inputs
    rate_of_change(h, r, density, TL, T, c, ROC); //write variables, don't declare types

    //formulas
    change_in_temperature = time_step * ROC;
    T = T + change_in_temperature;
    time = time + time_step;

    //logic test to break from loop and finish program
    if(abs(TL - T) <= 0.1)
    {
        //print out results
        cout << time << " \t\t " << r << endl;
        outfile << time << " \t\t " << r << endl;

        //break from loop
        d = 2;
    }
} while(d == 1);

cout << endl;

system("pause");
return 0;
}

```


//sub-function -- rate of change

```
void rate_of_change(double h, double r, double density, double TL, double T, double c, double &ROC)
```

```
{
```

```
    double pi = 4 * atan(1);
```

```
    double As = 4*pi*r*r; //surface area
```

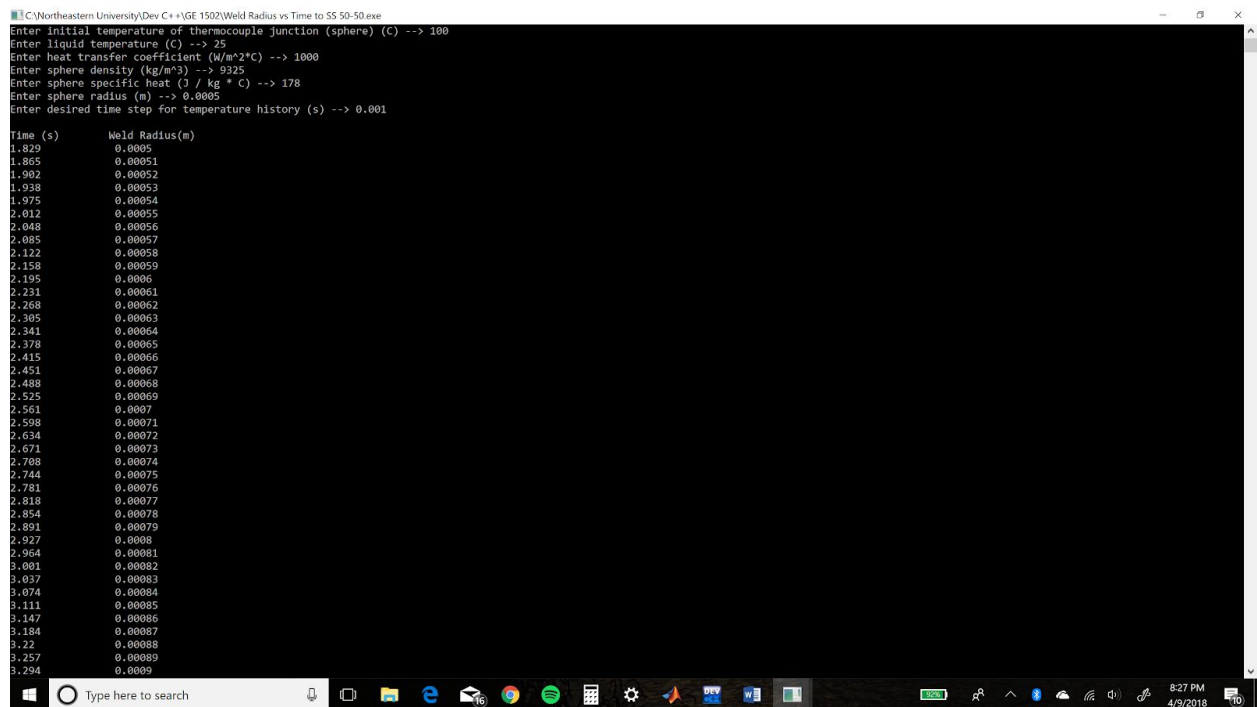
```
    double volume = pi*r*r*r*(4.0/3.0); //volume
```

```
    double m = density * volume;
```

```
    ROC = (h*As*(TL - T) ) / (m*c);
```

```
}
```

C++ results



The screenshot shows a Windows command prompt window titled "C:\Northeastern University\Dev C++\GE 1502\Weld Radius vs Time to SS 50-50.exe". The program prompts for several input values: Initial temperature of thermocouple junction (sphere) (C) --> 100, Enter liquid temperature (C) --> 25, Enter heat transfer coefficient (W/m^2*C) --> 1000, Enter sphere density (kg/m^3) --> 9325, Enter sphere specific heat (J / kg * C) --> 178, Enter sphere radius (m) --> 0.0005, and Enter desired time step for temperature history (s) --> 0.001. Below the prompts, a table is displayed with two columns: "Time (s)" and "Weld Radius(m)". The table contains 30 rows of data, showing a steady increase in both time and weld radius over the 3.294-second interval.

Time (s)	Weld Radius(m)
1.829	0.0005
1.865	0.00051
1.902	0.00052
1.938	0.00053
1.975	0.00054
2.012	0.00055
2.048	0.00056
2.085	0.00057
2.122	0.00058
2.158	0.00059
2.195	0.0006
2.231	0.00061
2.268	0.00062
2.305	0.00063
2.341	0.00064
2.378	0.00065
2.415	0.00066
2.451	0.00067
2.488	0.00068
2.525	0.00069
2.561	0.0007
2.598	0.00071
2.634	0.00072
2.671	0.00073
2.708	0.00074
2.744	0.00075
2.781	0.00076
2.818	0.00077
2.854	0.00078
2.891	0.00079
2.927	0.0008
2.964	0.00081
3.001	0.00082
3.037	0.00083
3.074	0.00084
3.111	0.00085
3.147	0.00086
3.184	0.00087
3.22	0.00088
3.257	0.00089
3.294	0.0009

```
C:\Northeastern University\Dev C++\GE 1502\Weld Radius vs Time to SS 50-50.exe
3.33      0.00091
3.367     0.00092
3.404     0.00093
3.44      0.00094
3.477     0.00095
3.513     0.00096
3.55      0.00097
3.587     0.00098
3.623     0.00099
3.66      0.001
3.697     0.00101
3.733     0.00102
3.77      0.00103
3.806     0.00104
3.843     0.00105
3.88      0.00106
3.916     0.00107
3.953     0.00108
3.99      0.00109
4.026     0.0011
4.063     0.00111
4.099     0.00112
4.136     0.00113
4.173     0.00114
4.209     0.00115
4.246     0.00116
4.283     0.00117
4.319     0.00118
4.356     0.00119
4.393     0.0012
4.429     0.00121
4.466     0.00122
4.502     0.00123
4.539     0.00124
4.576     0.00125
4.612     0.00126
4.649     0.00127
4.686     0.00128
4.722     0.00129
4.759     0.0013
4.795     0.00131
4.832     0.00132
4.869     0.00133
4.905     0.00134
4.942     0.00135
4.979     0.00136
5.015     0.00137
Press any key to continue . . .
```

MATLAB m-file

```
fid = fopen('weld_radius_time_60-40.txt', 'r');
```

```
formatSpec = '%f';
```

```
s = [2 Inf];
```

```
array = fscanf(fid, formatSpec, s);
```

```
length = max(size(array));
```

```
for i = 1 : length
```

```
    time1(i) = array(1, i);
```

```
    radius1(i) = array(2, i);
```

```
end
```

```
plot(time1, radius1)
```

```
xlabel('Time (s)');
```

```
ylabel('Weld Radius (m)');
```

```
title('Weld Radius vs Time to Steady State Temperature');
```

```
hold on
```

```
fid = fopen('weld_radius_time_50-50.txt', 'r');
```

```
formatSpec = '%f';
```

```

s = [2 Inf];

array = fscanf(fid, formatSpec, s);
length = max(size(array));

for i = 1 : length
    time2(i) = array(1, i);
    radius2(i) = array(2, i);
end

plot(time2, radius2);
hold off

```

Combined MATLAB plot

Note: **Red** = Tin / Lead 50-50, **Blue** = Tin / Lead 60-40

