Final Assignment: MATLAB and C++

Jordan Lian

GE 1502 -- 308 Hurtig Hall

10:30 - 11:35am

Professor Whalen

Table of Contents

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

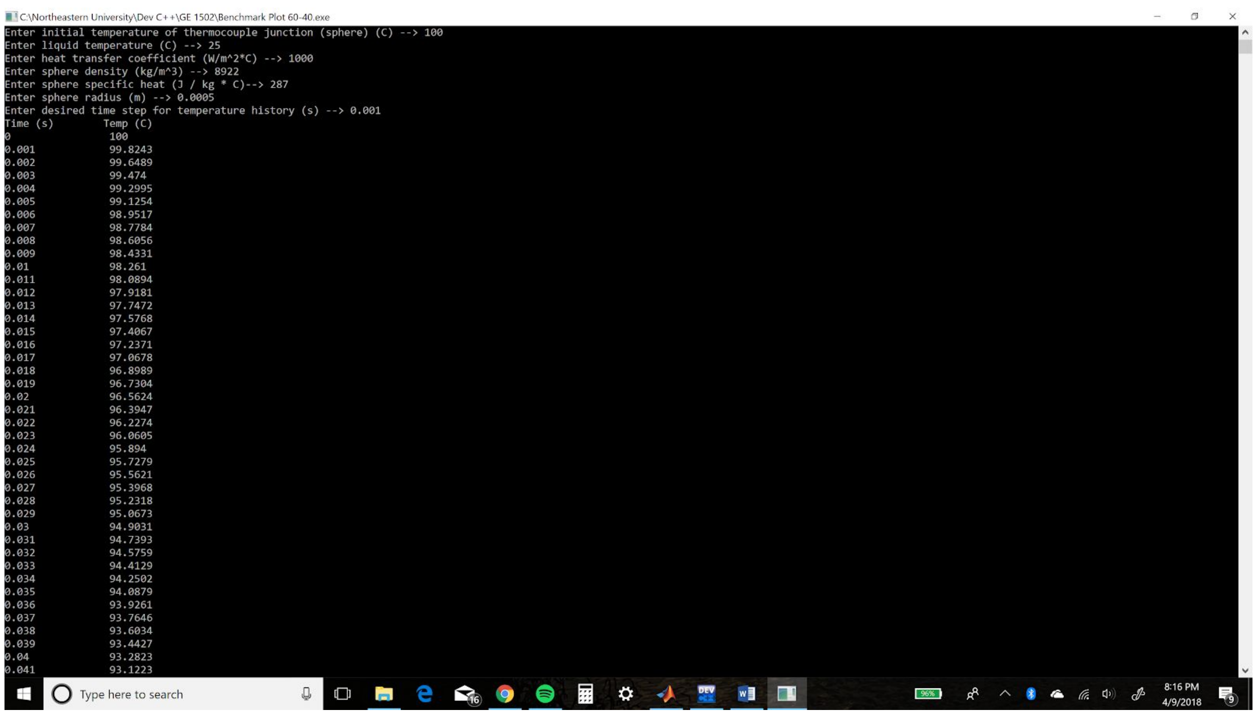
C++ source code

```
#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) --> ";
```

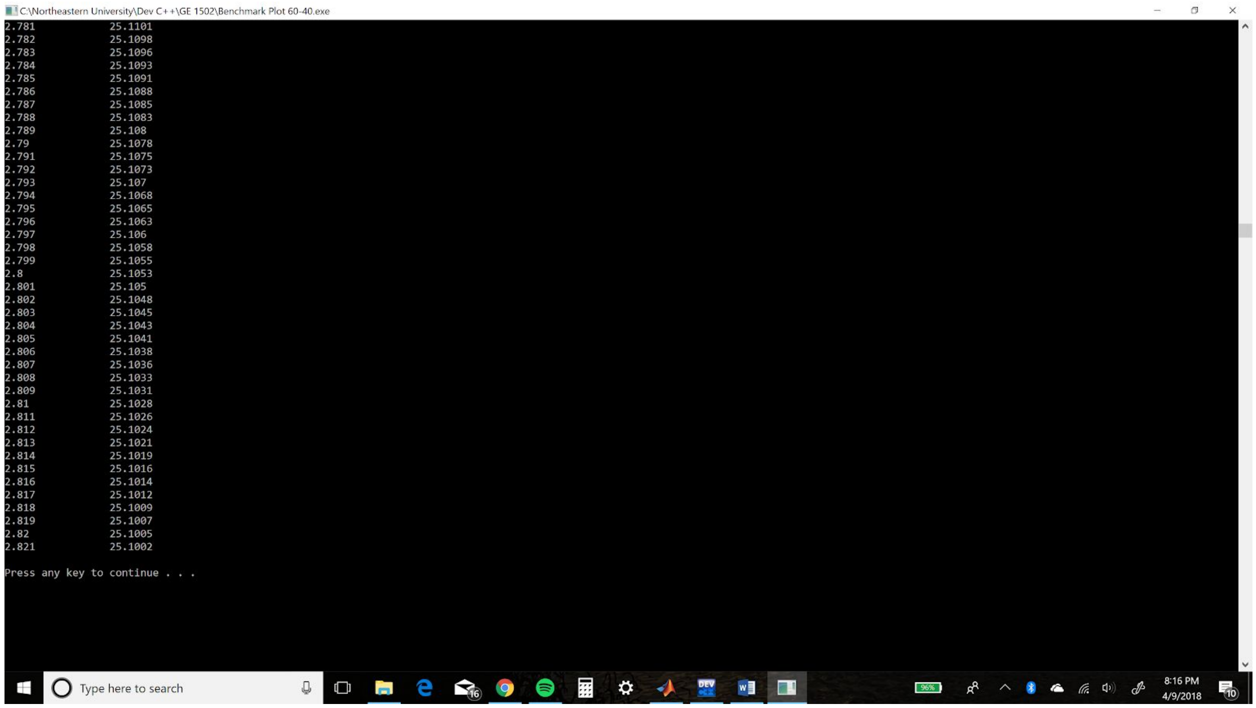
```
cin >> TI;

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:\Northeastern 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^2*C) --> 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
```



Tin / Lead 50/50

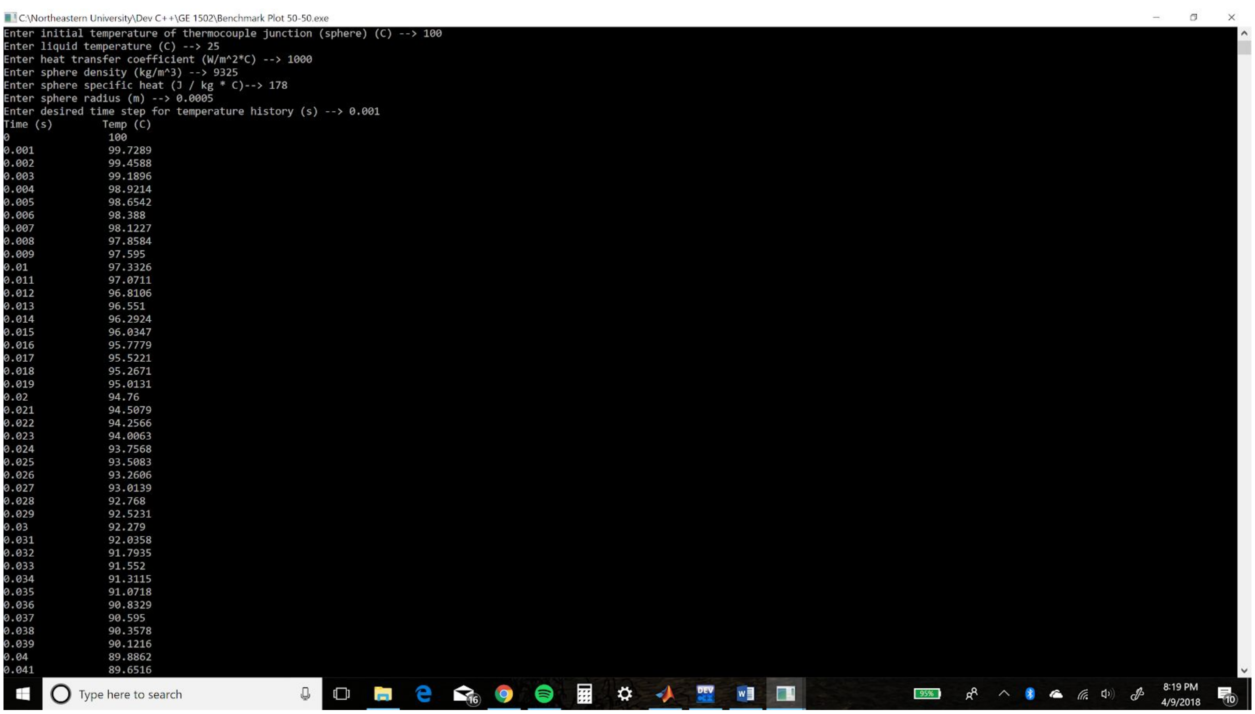
C++ source code

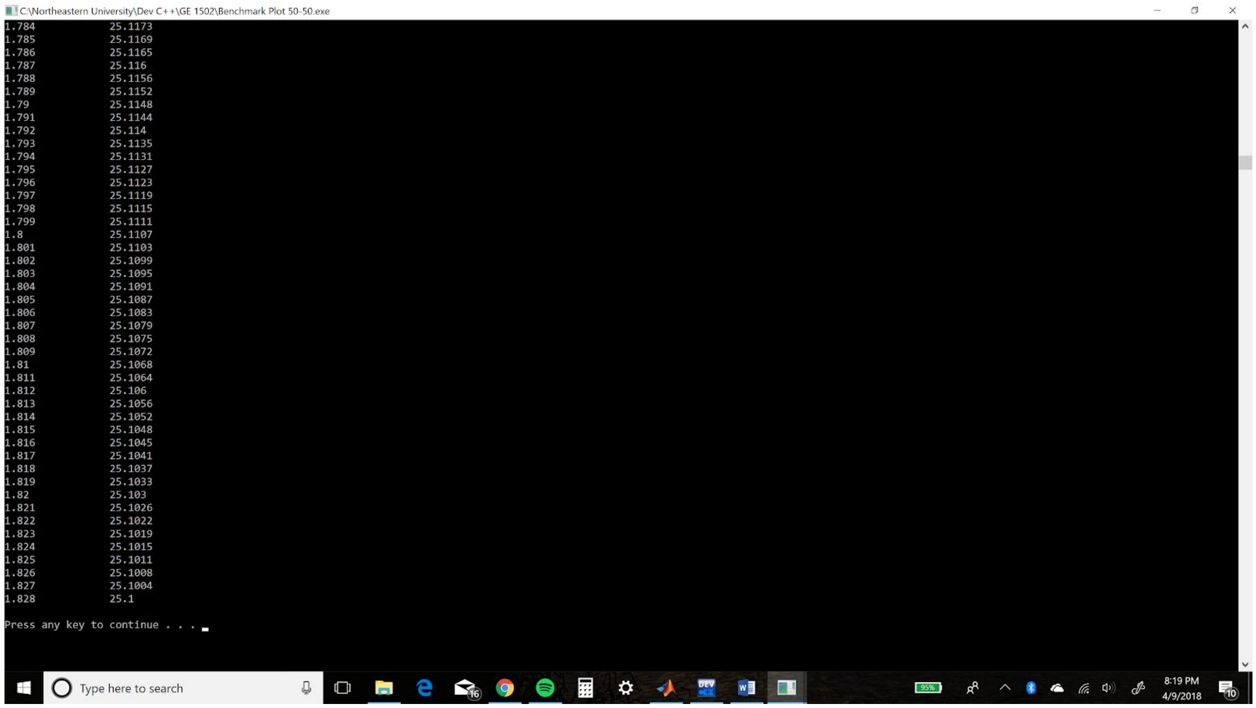
```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>

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





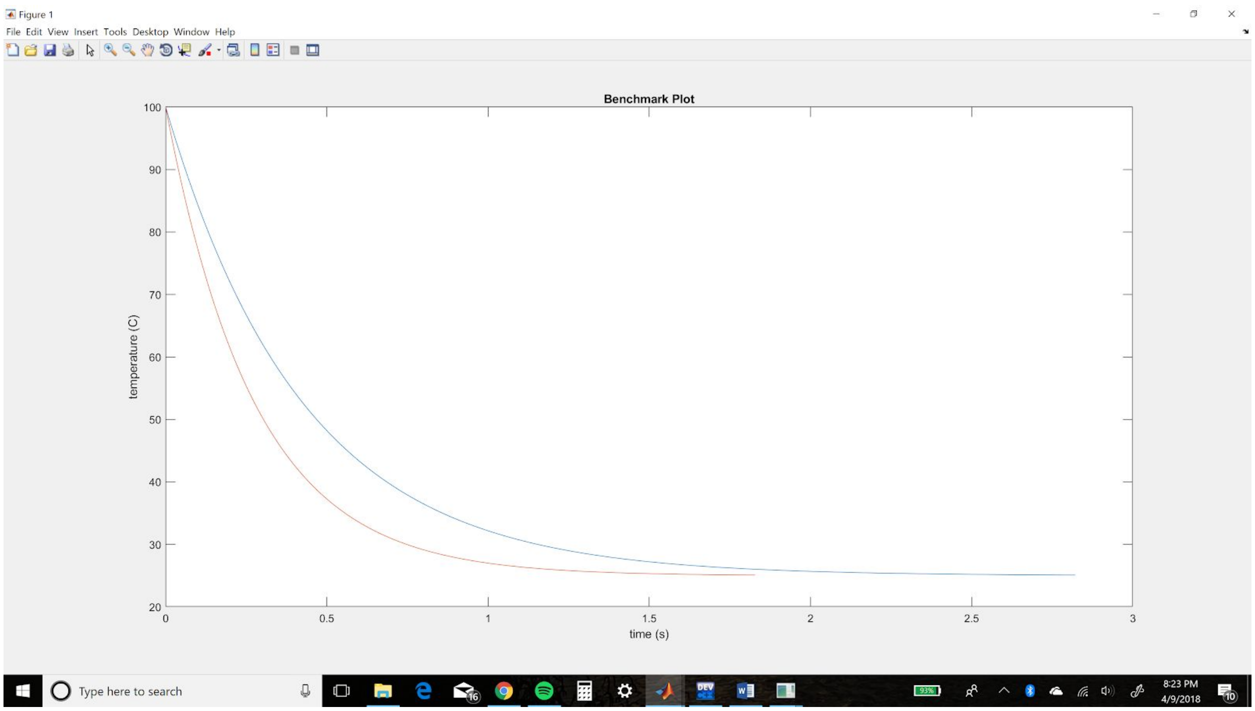
MATLAB m-file

```
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

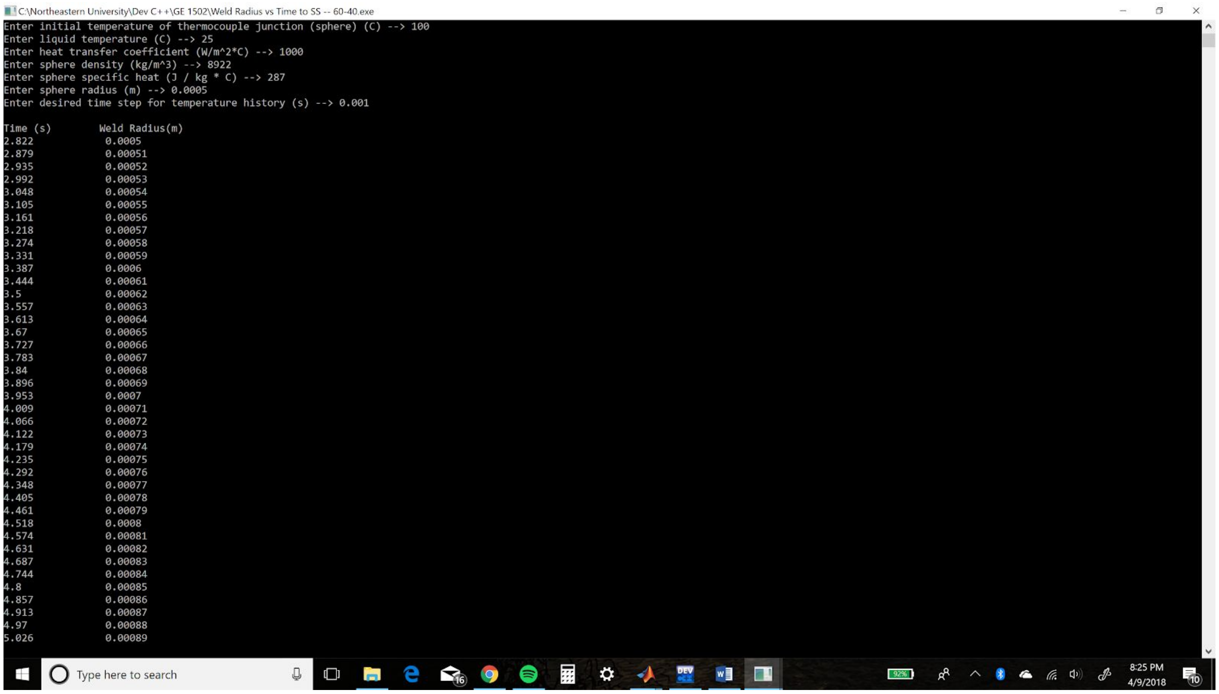
```
#include <iostream>
#include <cstdlib>
#include <cmath>
```

```
#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


```
C:\Northeastern University\Dev C++\GE 1502\Weld Radius vs Time to SS -- 60-40.exe
Enter 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) --> 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)        Weld Radius(m)
2.822            0.0005
2.879            0.00051
2.935            0.00052
2.992            0.00053
3.048            0.00054
3.105            0.00055
3.161            0.00056
3.218            0.00057
3.274            0.00058
3.331            0.00059
3.387            0.0006
3.444            0.00061
3.5              0.00062
3.557            0.00063
3.613            0.00064
3.67             0.00065
3.727            0.00066
3.783            0.00067
3.84             0.00068
3.896            0.00069
3.953            0.0007
4.009            0.00071
4.066            0.00072
4.122            0.00073
4.179            0.00074
4.235            0.00075
4.292            0.00076
4.348            0.00077
4.405            0.00078
4.461            0.00079
4.518            0.0008
4.574            0.00081
4.631            0.00082
4.687            0.00083
4.744            0.00084
4.8              0.00085
4.857            0.00086
4.913            0.00087
4.97             0.00088
5.026            0.00089
```


Tin / Lead 50/50

C++ source code

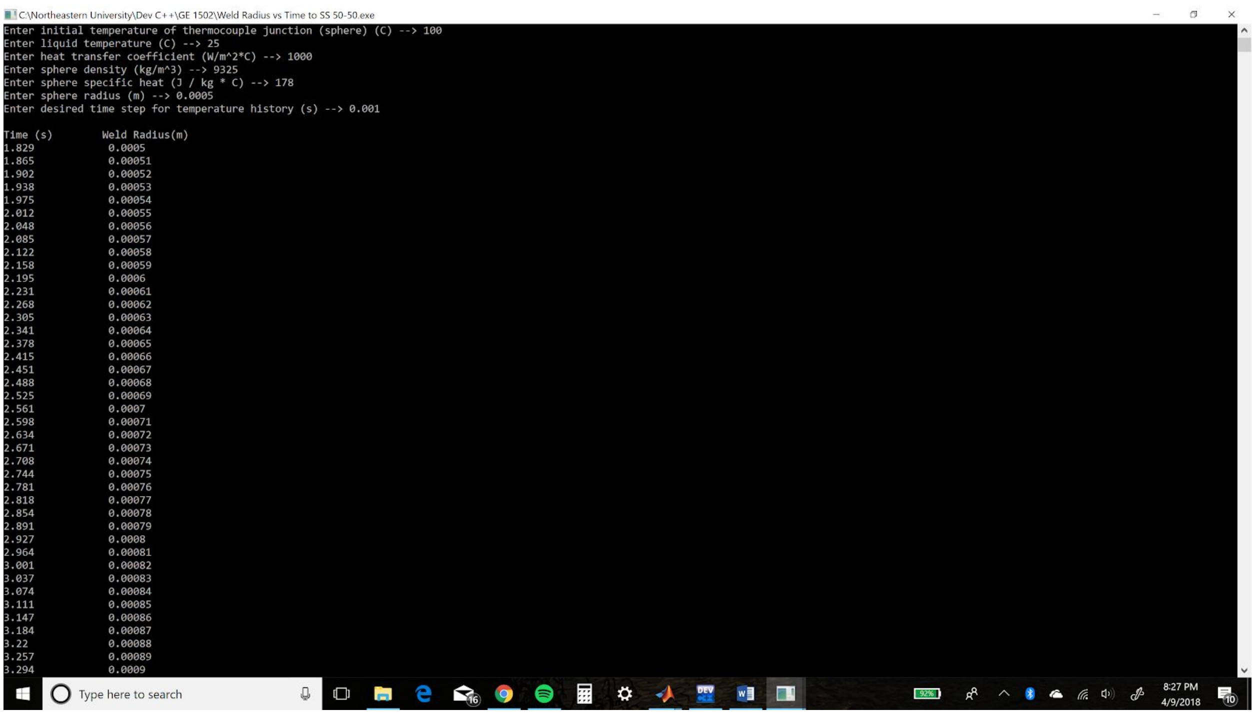
```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>

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;
```

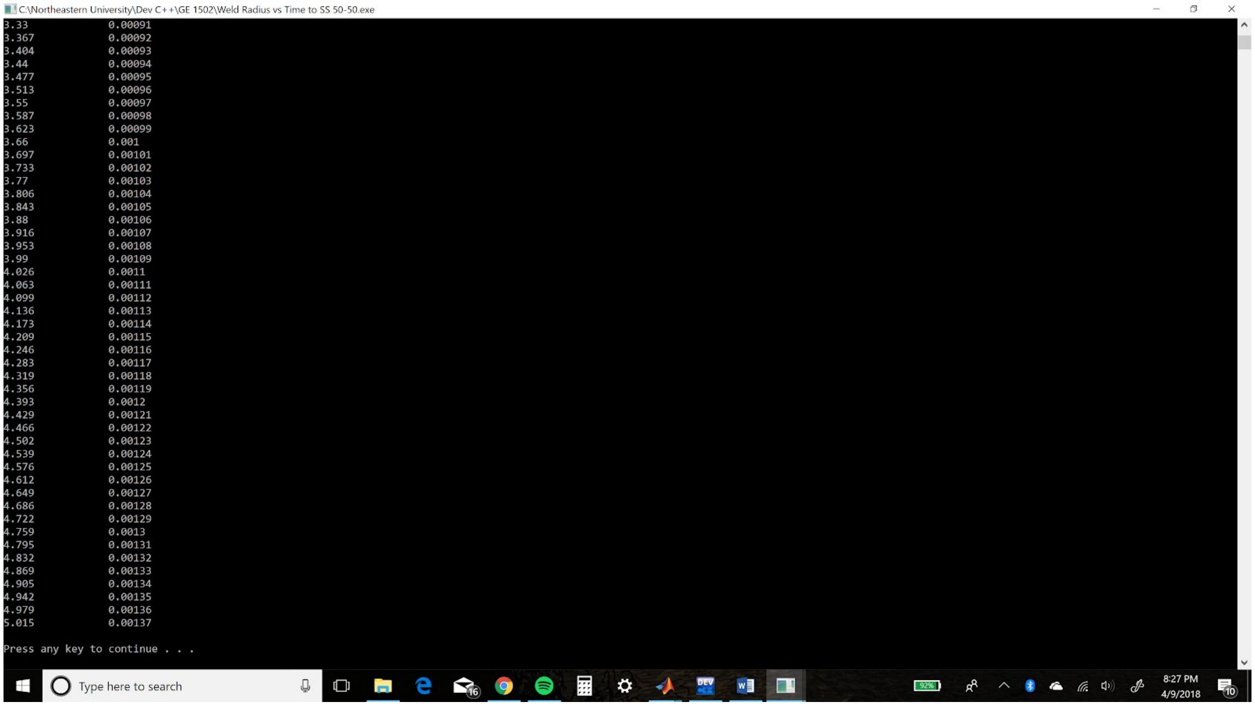
```
//output file for results
ofstream outfile;
outfile.open("weld_radius_time_50-50.txt"); //60-40 txt file

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





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