

```

#include <iostream>
#include <discpp.h>
#include <FileHandle.hpp>

#include <vector>
#include <string>
#include <sstream>
#include <limits>
#include <algorithm>
#include <tuple>

using namespace std;

static Dislin dlin;

#define k_B 1.3806488E-23
#define sigma_B 5.670373E-8
#define m_e 9.10938291E-31 //kg
#define m_p 1.67262178E-27 //kg
#define u 1.66053873E-27 //kg
#define m_H (1.00794*u) //kg
#define eV 1.60217657E-19 //joules
#define h 6.62606957E-34
#define G 6.673E-11
#define hbar 1.054571596E-34
#define pi 3.141592654
#define c_0 299792458.0
#define q_e 1.602176462E-19
#define H_0 (70.0*1000.0/3.08567758E22)
#define GYR (365.25*24.0*60.0*60.0*1E9)
#define AU 149597870700.0 //meters
#define pc (206264.81 * AU)
#define M_Sun 1.989E30 //kg
#define R_Sun 6.958E8 //meters
#define L_Sun 3.846E26 //watts

const double GYRH_0 = GYR*H_0;
char legendText[255] = { '/0' };
int legendIndex = 1;

void dislinInit(char* xName, char* yName, char* title){
    dlin.disini();
    //dlin.nochek();
    dlin.pagera();
    dlin.name(xName, "X");
    dlin.name(yName, "Y");
    dlin.titlin(title, 1);

    legendIndex = 1;
    dlin.complx();
    dlin.texmod("ON");
};
//=====
void q3(){
    //=====PART E=====
    dislinInit("$x$", "$\\theta$", "Question 3e");

    dlin.graf(0,10,0,1,0,1,0,0.1);

    dlin.title();

    vector<double> v_x, v_theta;
    double delta_x = 0.00001;

    auto part_e_helper = [&](double n){
        v_x.clear();
        v_theta.clear();

        double t = 1.0, x = 0, dt = 0, ddt = 0;

```

```

bool curved = false;
int offset = 0;

v_x.push_back(x);
v_theta.push_back(t);

while (t >= 0 && x <= 10){
    x += delta_x;
    ddt = -(pow(t, n) + 2.0 / x * dt);
    dt += ddt*delta_x;
    t += dt*delta_x;

    v_x.push_back(x);
    v_theta.push_back(t);

    if (t <= 0.4 && !curved){
        curved = true;
        dlin.curve(v_x.data(), v_theta.data(), v_x.size());
        offset = v_x.size() - 1;
    }
}

if (t <= 0){
    cout << "n = " << n << endl;
    cout << "\t" << "t ~ " << t << " at x ~ " << x << endl;
    cout << "\t" << "dt ~ " << dt << " at x ~ " << x << endl;
    cout << "\t" << "ddt ~ " << dt << " at x ~ " << x << endl;
}

dlin.labels("conlab", "contun");
stringstream ss;
ss << "n = " << int(n);

dlin.congap(0.1);
dlin.labdis(800, "CONTUR");
dlin.conlab(ss.str().c_str());
//dlin.concrv(v_x.data(), v_theta.data(), v_x.size(), n);
dlin.concrv(&v_x[offset], &v_theta[offset], v_x.size() - offset, n);
};

part_e_helper(0);
part_e_helper(1);
part_e_helper(2);
part_e_helper(4);
part_e_helper(5);
part_e_helper(3);

//=====PART H=====

dlin.disfin();
dislinInit("$\\frac{r}{R_{star}}$", "Density $\\rho$ $(\\frac{kg}{m^3})$ $\\times 10^4$",
"Question 3h - Density");
dlin.labdig(-1, "v");
dlin.graf(0, 1, 0, 0.1, 0, 8, 0, 1);
dlin.nochek();
dlin.title();

vector<double> v_rho, v_r_Rstar;

double rho_central = 7.634E4; // from part g
double alpha = 1.009E8; //from part g

for(int i = 0; i < v_x.size(); i++){
    v_rho.push_back(pow(v_theta[i],3)*rho_central / 1E4);
    v_r_Rstar.push_back(v_x[i] / R_Sun*alpha);
}

dlin.curve(v_r_Rstar.data(), v_rho.data(), v_r_Rstar.size());

```

```

dlin.disfin();

dislinInit("$\\frac{r}{R_{star}}$", "Pressure $P$ $(Pa)$ $\\times 10^{15}$", "Question 3h -
Pressure");
dlin.labdig(-1, "Y");
dlin.graf(0, 1, 0, 0.1, 0, 13, 0, 1);
dlin.nochek();
dlin.title();

vector<double> v_P;
double K = 3.841E9;

for (int i = 0; i < v_rho.size(); i++){
    v_P.push_back(K * pow(v_rho[i] * 1E4, 4.0 / 3.0) / 1E15);
}

dlin.curve(v_r_Rstar.data(), v_P.data(), v_r_Rstar.size());
dlin.disfin();

dislinInit("$\\frac{r}{R_{star}}$", "Temperature $T$ $(K)$ $\\times 10^6$", "Question 3h -
Temperature");
dlin.labdig(-1, "Y");
dlin.graf(0, 1, 0, 0.1, 0, 14, 0, 1);
dlin.nochek();
dlin.title();

vector<double> v_T;
double inv_mu = 2.0 * 0.55 + 3.0 / 4.0 * 0.4 + 0.5 * 0.05;

for (int i = 0; i < v_rho.size(); i++){
    v_T.push_back(v_P[i] * 1E15 / inv_mu / (v_rho[i]*1E4) / k_B*m_H / 1E6);
}

dlin.curve(v_r_Rstar.data(), v_T.data(), v_r_Rstar.size());
dlin.disfin();

//=====PART I=====

dislinInit("$\\frac{r}{R_{star}}$", "$\\frac{dL}{dr}$ $\\times 10^{18}$", "Question 3i");
dlin.graf(0, 1, 0, 0.1, 0, 2.1, 0, 0.2);
dlin.addlab("2.1", 2.1, 1, "Y");
dlin.nochek();
dlin.title();

vector<double> v_dL_dr;
double L = 0;

for (int i = 0; i < v_r_Rstar.size(); i++){
    double e_pp = 1.07E-7*v_rho[i]/1E1*0.55*0.55*pow(v_T[i],4);
    double e_cno = 8.24E-26*v_rho[i] / 1E1*0.55*0.55*0.03*pow(v_T[i], 19.9);
    double dL_dr = 4 * pi*pow(v_r_Rstar[i] * R_Sun, 2)*v_rho[i] * 1E4*(e_pp + e_cno) / 1E18;

    if (!isnan(dL_dr))
        L += dL_dr * delta_x * alpha;

    v_dL_dr.push_back(dL_dr);
}

cout << "Total Luminosity: " << L * 1E18 << " W = " << L * 1E18 / L_Sun << "L_Sun" << endl;
dlin.curve(v_r_Rstar.data(), v_dL_dr.data(), v_r_Rstar.size());
dlin.disfin();

auto part_g_helper = [&inv_mu, &v_x, &v_theta, &delta_x](double M, double R){
    double alpha_g = R*R_Sun / 6.897;
    double rho_central_g = -M_Sun*M / 4.0 / pi / pow(alpha_g, 3) / 6.897 / 6.897 / (-
0.0424291);
    double K_g = alpha_g*alpha_g*4.0*pi*G*(4.0 / 3.0 - 1.0) / 4.0*3.0 / pow(rho_central_g, 4.0
/ 3.0 - 2.0);
    double P_g = K_g * pow(rho_central_g, 4.0 / 3.0);

```

```

double T_g = P_g / inv_mu*m_H / rho_central_g / k_B;
double L_g = 0;

for (int i = 0; i < v_x.size(); i++){
    double rho = pow(v_theta[i], 3)*rho_central_g;
    double P = K_g * pow(rho, 4.0 / 3.0);
    double T = P / inv_mu*m_H / rho / k_B;
    double e_pp = 1.07E-7*(rho/1E5)*0.55*0.55*pow(T/1E6, 4);
    double e_cno = 8.24E-26*(rho/1E5)*0.55*0.55*0.03*pow(T/1E6, 19.9);
    double dL_dr = 4.0 * pi*pow((v_x[i]*alpha_g), 2)*rho*(e_pp + e_cno);

    if (!isnan(dL_dr))
        L_g += dL_dr * delta_x * alpha_g;
}

cout << "M = " << M << ", R = " << R << " : " << endl << "\t rho_central = " <<
rho_central_g << endl << "\t Temp = " << T_g << endl << "\t L = " << L_g / L_Sun << "L_Sun" << endl;
};

part_g_helper(0.5, 0.6);
part_g_helper(1.0, 1.0);
part_g_helper(20, 10);
}

int main() {
    dlin.metafl("PDF");
    dlin.imgfmt("RGB");
    dlin.filmod("COUNT");
    dlin.scrmod("REVERS");
#ifdef _DEBUG
    dlin.metafl("XWIN");
#endif

    q3();

    system("PAUSE");
    return(0);
}

```