Komparasi Metode Interpolasi Natural Cubic Spline dengan Clamped Cubic Spline

Muhammad Indra N. S. - 23515019
Program Magister Informatika
Institute Teknologi Bandung
Bandung, Indonesia
23515019@std.stei.itb.ac.id

***Abstract*—Salah satu pembahasan pada metode lanjut adalah tentang interpolasi. Interpolasi banyak digunakan ketika kita memiliki data sample atau hasil dari eksperimen dan menaksirkan nilai-nilai tersebut untuk mendapatkan nilai yang lebih baru. Cubic spline adalah salah satu metode yang digunakan untuk interpolasi. Paper ini bertujuan untuk membandingkan dua metode yang ada pada Cubic Spline, yaitu metode Natural Cubic Spline dengan metode Clamped Cubic Spline untuk kasus penyelesaian interpolasi.**

Keywords—interpolasi; cubic spline; natural cubic spline; clamped cubic spline

I. PENDAHULUAN

Matematika merupakan salah satu bidang ilmu yang dekat dengan kehidupan manusia sehari-hari. Banyak persoalan di dunia nyata yang dapat direpresentasikan dalam bentuk persoalan matematika. Persoalan-persoalan tersebut dapat ditemui dalam berbagai bidang, misalnya bidang rekayasa.

Pada umumnya persoalan matematika diselesaikan dengan menggunakan rumus-rumus yang sudah lazim digunakan. Cara formal untuk menyelesaikan persoalan matematika seperti ini disebut juga metode analitik atau metode sejati. Solusi yang didapatkan dengan menggunakan metode analitik merupakan solusi yang sejati atau memiliki nilai galat sama dengan nol. Sayangnya, banyak bentuk persoalan matematika yang tidak dapat diselesaikan hanya menggunakan metode analitik. Oleh karena itu, manusia mencari cara untuk mencari solusi dari persoalan-persoalan yang tidak dapat diselesaikan dengan mengunakan metode analitik. Solusinya didefinisikan sebagai metode numerik.

Dengan menggunakan metode numerik, persoalan-persoalan yang tidak dapat diselesaikan dengan menggunakan metode analitik dapat dicari solusi hampirannya. Solusi hampiran ini tentu saja memiliki selisih perbedaan nilai dengan solusi sejati. Nilai selisih ini disebut galat. Ada beberapa jenis persoalan matematika yang dapat diselesaikan dengan menggunakan metode numerik. Salah satunya adalah persoalan interpolasi polinom. Persoalan interpolasi polinom merupakan persoalan untuk mencari suatu polinom yang dapat menghubungkan sekumpulan titik yang diketahui. Polinom yang didapatkan dari interpolasi ini dapat digunakan sebagai alat bantu untuk mencari katian yang hilang. Interpolasi polinom sering digunakan untuk membantu menggambar kurva dari suatu hasil percobaan. Ada beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan interpolasi polinom. Salah satu bentuk penyelesaian interpolasi polinom adalah dengan menggunakan metode Interpolasi Cubic Spline.

Paper ini terbagi menjadi beberapa section. Section 2 akan membahas mengenai Interpolasi Cubic Spline. Section 3 dan 4 akan membahas mengenai Natural Cubic Spline dan Clamped Cubic Spline, disertai dengan algoritmanya. Pada section 5 akan mengimplementasikan kedua algoritma dalam bahasa C, dan membandingkan antar keduanya.

II. INTERPOLATION CUBIC SPLINE

Interpolasi spline merupakan metode yang memiliki perbedaan cukup signifikan dengan metode interpolasi lainnya, seperti polinom Lagrange dan Newton. Pada metode lain, polinom interpolasi yang didapatkan berjumlah satu buah dan menghubungkan seluruh titik data. Sementara itu pada interpolasi spline, titik data yang berdekatan dihubungkan dengan satu polinom sehingga jumlah polinom interpolasi yang didapatkan adalah sejumlah m-1 (dengan m adalah jumlah titik data).

Seperti pada metode lainnya, orde polinom untuk interpolasi dapat ditentukan. Harapannya adalah semakin tinggi orde polinom, maka kurva yang dihasilkan akan semakin mulus dan mendekati kurva asli. Pada interpolasi spline, orde yang sering digunakan adalah orde tiga. Metode interpolasi spline dengan menggunakan polinom orde tiga disebut juga metode cubic spline. Dalam fungsi cubic spline, setiap pasang titip di interpolasi dengan polinom kubik sebagai berikut :

$$S_k(x) = a_k x^3 + b_k x^2 + c_k x + d_k \tag{1}$$

untuk k = 1,2,…,n dan n adalah jumlah titik data dikurangi satu atau dengan kata lain jumlah polinom kubik.

Setiap polinom kubik $S_k(x)$ memiliki empat buah tetapan yang tidak diketahui (a_k, b_k, c_k, d_k). Karena terdapat n buah polinom kubik berarti ada 4n tetapan yang harus dipecahkan dengan 4n persamaan. Keseluruhan persamaan tersebut diperoleh dari lima persyaratan berikut :

1) Nilai fungsi harus sama pada titik simpul (2n–2 buah persamaan).

2) Fungsi kubik yang pertama dan terakhir melalui titik titik ujung (2 buah persamaan).

3) Turunan pertama pada titik simpul harus sama (n–1 buah persamaan).

4) Turunan kedua pada titik simpul harus sama (n–1 buah persamaan).

5) Turunan kedua pada titik-titik ujung adalah nol (2 buah persamaan).

Algoritma Interpolation Cubic Spline :

1) Membuat sistem persamaan linear untuk menghitung turunan S kedua.

2) Mencari selang yang relevan terhadap titik interpolasi yang diinginkan. Sub selang yang relevan adalah apabila titik interpolasi terdapat di antara subselang tersebut.

3) Bila sub selang relevan adalah sub selang ke-n, hitung nilai S.

II. Natural Cubic Spline

Natural Cubic Spline adalah suatu kondisi pada saat menetukan batasan nilai S berdasarkan cubic spline. Batasan yang ditentukan pada Natural Cubic Spline adalah turunan kedua dari S bernilai 0,

$$S''(x_0) = S''(x_n) = 0.$$

Jadi ketika ingin membuat interpolasi cubic spline S untuk fungsi f, definisikan nilai angka $x_0 < x_1 < \ldots < x_n$, untuk memenuhi nilai $S''(x_0) = S''(x_n) = 0$.

Algoritma untuk Natural Cubic Spline

1) *Untuk setiap i = 0, 1 sampai n-1, tentukan nilai untuk* $h_i = x_{i+1} - x_i$.
2) *Untuk setiap i = 1, 2 sampai n-1, tentukan nilai untuk* $\alpha_i = 3/h_i (\alpha_{i+1} - \alpha_i) - 3/h_{i-1} (\alpha_i - \alpha_{i-1})$.
3) *Tentukan nilai* $l_0 = 1$; $\mu_0 = 0$; $z_0 = 0$.
4) *Untuk setiap i = 1, 2 sampai n-1, tentukan nilai untuk* $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$; $\mu_i = h_i / l_i$; $z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$.
5) *Tentukan nilai* $l_n = 1$; $z_n = 0$; $c_n = 0$.
6) *Untuk setiap j = n-1, n-2 sampai 0, tentukan nilai untuk* $c_j = z_j - \mu_j c_{j+1}$; $b_j = (a_{j+1} - a_j) / h_j - h_j (c_{j+1} + 2c_j) / 3$; $d_j = (c_{j+1} - c_j) / (3h_j)$.

III. Clamped Cubic Spline

Clamped Cubic Spline adalah suatu kondisi pada saat menetukan batasan nilai S berdasarkan cubic spline. Batasan yang ditentukan pada Clamped Cubic Spline adalah turunan pertama dari S bernilai turunan pertama dari fungsi x,

$$S'(x_0) = f'(x_0) \text{ dan } S'(x_n) = f'(x_n).$$

Jadi ketika ingin membuat interpolasi cubic spline S untuk fungsi f, definisikan nilai angka $x_0 < x_1 < \ldots < x_n$, untuk memenuhi nilai $S'(x_0) = f'(x_0)$ *dan* $S'(x_n) = f'(x_n)$.

Algoritma untuk Clamped Cubic Spline

1) *Untuk setiap i = 0, 1 sampai n-1, tentukan nilai untuk* $h_i = x_{i+1} - x_i$.
2) *Tentukan nilai untuk* $\alpha_0 = 3(\alpha_1 - \alpha_0)/h_0 - 3FPO$, *FPO adalah turunan pertama fungsi f terhadap* x_0.
3) *Untuk setiap i = 1, 2 sampai n-1, tentukan nilai* $\alpha_i = 3/h_i (\alpha_{i+1} - \alpha_i) - 3/h_{i-1} (\alpha_i - \alpha_{i-1})$.
4) *Tentukan nilai* $l_0 = 2h_0$; $\mu_0 = 0.5$; $z_0 = \alpha_0 / l_0$.
5) *Untuk setiap i = 1, 2 sampai n-1, tentukan nilai untuk* $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$; $\mu_i = h_i / l_i$; $z_i = (\alpha_i - h_{i-1} z_{i-1}) / l_i$.
6) *Tentukan nilai* $l_n = h_{n-1}(2 - \mu_{n-1})$; $z_n = (\alpha_n - h_{n-1}\ {}_{n-1}) / l_n$; $c_n = z_n$.
7) *Untuk setiap j = n-1, n-2 sampai 0, tentukan nilai untuk* $c_j = z_j - \mu_j c_{j+1}$; $b_j = (a_{j+1} - a_j) / h_j - h_j (c_{j+1} + 2c_j) / 3$; $d_j = (c_{j+1} - c_j) / (3h_j)$

IV. Implementasi

Bahasa yang digunakan dalam pengerjaan program ini adalah bahasa C++. Untuk metode Natural Cubic Spline program lengkap seperti di bawah ini:

```
#include <iostream>
#include <fstream>
#include <math.h>
using namespace std;

int n = 20, i, j;
double x[] =
{0.9,1.3,1.9,2.1,2.6,3.0,3.9,4.4,4.7,
5.0,6.0,7.0,8.0,9.2,10.5,11.3,11.6,
12.0,12.6,13.0,13.3};
double fx[] =
{1.3,1.5,1.85,2.1,2.6,2.7,2.4,2.15,
2.05,2.1,2.25,2.3,2.25,1.95,1.4,0.9,
0.7,0.6,0.5,0.4,0.25};
double h[100], alp[100], l[100],
miu[100];
double c[100], b[100], d[100], z[100];

double f(double a, double b, double c,
double d, double x0, double xn){
    return a + b*(x0-xn) + c*pow((x0-
xn),2) + d*pow((x0-xn),3);
}
```

```
void save_file(){
// Simpan data asli
ofstream out1;
out1.open("data-asli.txt");
for(i=0; i<n; i++)
    out1 << x[i] << " " << fx[i] <<
endl;     out1.close();
double p = x[0];
double q = x[n];
double y = fx[0];
int maks = 200;
double dx = (q-p)/maks;
// Simpan data natural cubic spline
ofstream out2;
out2.open("data-natural-spline.txt");
out2 << p << " " << y << endl;
for(i=0; i<maks; i++){
         p = p + dx;
         for(j=0; j<n; j++){
             if(p>=x[j] && p<x[j+1]){
                 y = f(fx[j], b[j],
c[j], d[j], p, x[j]);
                 out2 << p << " " << y
<< endl;
             }
         }
     }
out2.close();
// Simpan koefisien natural cubic
spline
ofstream out3("koefisien-natural-
spline.txt");
for(i=0; i<n; i++){
         out3 << fx[i] << " " << b[i]
<< " " << c[i] << " " << d[i] << endl;
}
out3.close();
// Simpan fungsi piecewise natural
cubic spline
ofstream out4("fungsi-natural-
spline.txt");
for(i=0; i<n; i++){
         out4<<"S"<<i<<" = "<<fx[i]<<"
+ "<<b[i]<< "(x - "<<x[i]<<")+"<<c[i];
         out4<<"(x-"<<x[i]<<")^2 +
"<<d[i]<<"(x-"<< x[i]<<")^3"<<endl;
}
out4.close();
}

int main(){
 //Step 1
for(i=0; i<n; i++){
         h[i] = x[i+1]-x[i];
}
//Step 2
for(i=1; i<n; i++){
         alp[i] = (3./h[i])*(fx[i+1]-
fx[i])-(3./h[i-1])* (fx[i]-fx[i-1]);
}
//Step 3
l[0]=1.;
miu[0]=0.;
z[0]=0.;
```

```
//Step 4
for(i=1; i<n; i++){
         l[i] = 2.*(x[i+1]-x[i-1])-
h[i-1]*miu[i-1];
         miu[i] = h[i]/l[i];
         z[i] = (alp[i]-(h[i-1]*z[i-
1]))/l[i];
}
//Step 5
l[n]=1.;
z[n]=0.;
c[n]=0.;
//Step 6
for(j=n-1; j>=0; j--){
         c[j] = z[j]-miu[j]*c[j+1];
         b[j] = (fx[j+1]-fx[j])/h[j]-
h[j]*(c[j+1]+2.* c[j])/3.;
         d[j] = (c[j+1]-
c[j])/(3.*h[j]);
}
// Simpan file
save_file();
return 0;
}
```

Sedangkan untuk implementasi Clamped Cubic Spline

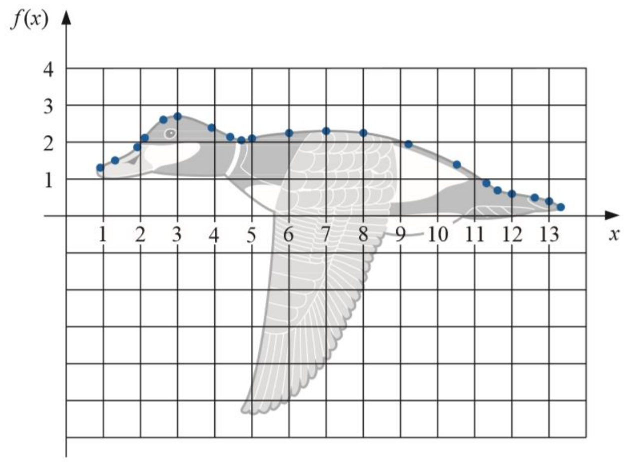
```
#include <iostream>
#include <fstream>
#include <math.h>
using namespace std;

int n = 20, i, j;
double x[] =
{0.9,1.3,1.9,2.1,2.6,3.0,3.9,4.4,4.7,
5.0,6.0,7.0,8.0,9.2,10.5,11.3,11.6,
12.0,12.6,13.0,13.3};
double fx[]= {1.3,1.5,1.85,2.1,2.6,
2.7,2.4,2.15,2.05,2.1,2.25,2.3,2.25,
1.95,1.4,0.9,0.7,0.6,0.5,0.4,0.25};
double h[100], alp[100], l[100],
miu[100];
double c[100], b[100], d[100], z[100];

double f(double a, double b, double c,
double d, double x0, double xn){
     return a + b*(x0-xn) + c*pow((x0-
xn),2) + d*pow((x0-xn),3);
}

void save_file(){
// Simpan data asli
ofstream out1;
out1.open("data-asli.txt");
for(i=0; i<=n; i++)
        out1<<x[i]<<" "<<fx[i] <<endl;
out1.close();

double p = x[0];
double q = x[n];
double y = fx[0];
int maks = 200;
double dx = (q-p)/maks;
```

```
// Simpan data natural cubic spline
ofstream out2;
out2.open("data-clampled-spline.txt");
out2 << p << " " << y << endl;
for(i=0; i<maks; i++){
      p = p + dx;
      for(j=0; j<n; j++){
           if(p>=x[j] && p<x[j+1]){
                y = f(fx[j],b[j],c[j],
d[j],p,x[j]);
                out2 << p << " " << y <<
endl;
           }
      }
}
out2.close();

// Simpan koefisien natural cubic
spline
ofstream out3("koefisien-clamped-
spline.txt");
for(i=0; i<n; i++){
        out3 << fx[i] << " " << b[i] <<
" " << c[i] << " " << d[i] << endl;
}
out3.close();

// Simpan fungsi piecewise natural
cubic spline
ofstream out4("fungsi-clamped-
spline.txt");
for(i=0; i<n; i++){
        out4<<"S"<<i<<" = "<<fx[i]<<" +
"<<b[i]<<"(x - "<<x[i]<<")+"<<c[i];
        out4<<"(x-"<<x[i]<<")^2 +
"<<d[i]<<"(x-"<<x[i]<<")^3"<<endl;
}
out4.close();
}

int main(){
double fpo, fpn;
cout << "Input f'(x0) = "; cin >> fpo;
cout << "Input f'(xn) = "; cin >> fpn;

//Step 1
for(i=0; i<n; i++){
h[i] = x[i+1]-x[i];
}

//Step 2
alp[0] = 3*(fx[1]-fx[0])/h[0]-3*fpo;
alp[n] = 3*fpn-3*(fx[n]-fx[n-1])/h[n-
1];

//Step 3
for(i=1; i<n; i++){
alp[i] = (3./h[i])*(fx[i+1]-fx[i])-
(3./h[i-1])*(fx[i]-fx[i-1]);
}

//Step 4
l[0]=2.*h[0]; miu[0]=0.5;
z[0]=alp[0]/l[0];
```

```
//Step 5
for(i=1; i<n; i++){
l[i] = 2.*(x[i+1]-x[i-1])-h[i-
1]*miu[i-1];
miu[i] = h[i]/l[i];
z[i] = (alp[i]-(h[i-1]*z[i-1]))/l[i];
}

//Step 6
l[n] = h[n-1]*(2.-miu[n-1]);
z[n] = (alp[n]-h[n-1]*z[n-1])/l[n];
c[n] = z[n];
//cout << h[n-2] << " " << z[n] << " "
<< c[n] << endl;

//Step 7
for(j=n-1; j>=0; j--){
c[j] = z[j]-miu[j]*c[j+1];
b[j] = (fx[j+1]-fx[j])/h[j]-
h[j]*(c[j+1]+2.*c[j])/3.;
d[j] = (c[j+1]-c[j])/(3.*h[j]);
}

//Simpan file
save_file();
return 0;
}
```

Untuk melakukan pengujian, di ambil sebuah contoh soal tentang penyelesaian interpolasi dengan menggunakan metode Natural Cubic Spline dan Clamped Cubic Spline.

Fig. 1. Contoh graphic yang akan di interpolasi



Berdasarkan dari titik-titik pada fig. 1.diketahui nilai x dan f(x) berturut-turut adalah sebagai berikut:

x : 0.9, 1.3, 1.9, 2.1, 2.6, 3.0, 3.9, 4.4, 4.7, 5.0, 6.0, 7.0, 8.0, 9.2, 10.5, 11.3, 11.6, 12.0, 12.6, 13.0, 13.3.

f(x) : 1.3, 1.5, 1.85, 2.1, 2.6, 2.7, 2.4, 2.15, 2.05, 2.1, 2.25, 2.3, 2.25, 1.95, 1.4, 0.9, 0.7, 0.6, 0.5, 0.4, 0.25.

Dengan menggunakan program yang telah dibuat, hasil yang diperoleh untuk kasus soal di atas dengan Natural Cubic Spline terlihat seperti pada fig. 2 di bawah ini:

Fig. 2. Hasil perhitungan koefisien Natural Cubic Spline

Koefisien Fungsi Interpolasi Natural Cubic Spline

a	b	c	d
1.3	0.539624	0	-0.247649
1.5	0.420752	-0.297179	0.946912
1.85	1.0868	1.40726	-2.95638
2.1	1.29494	-0.366567	-0.446635
2.6	0.593399	-1.03652	0.445051
2.7	-0.0221911	-0.502457	0.17416
2.4	-0.503406	-0.0322258	0.0780757
2.15	-0.477075	0.0848877	1.31417
2.05	-0.0713162	1.26764	-1.58122
2.1	0.26234	-0.155455	0.0431153
2.25	0.0807755	-0.0261092	-0.00466634
2.3	0.0145582	-0.0401082	-0.02445
2.25	-0.139008	-0.113458	0.0174707
1.95	-0.335834	-0.0505636	-0.0127279
1.4	-0.53183	-0.100202	-0.0203252
0.9	-0.731178	-0.148983	1.21341
0.7	-0.492949	0.943082	-0.839275
0.6	-0.141335	-0.0640482	0.0363821
0.5	-0.1789	0.00143957	-0.447971
0.4	-0.392775	-0.536126	0.595695

Untuk menggambarkan plot dari hasil perhitungan program Natural Cubic Spline digunakan library tambahan dari gnuplot. Sehingga menghasilkan plot seperti pada fig. 4

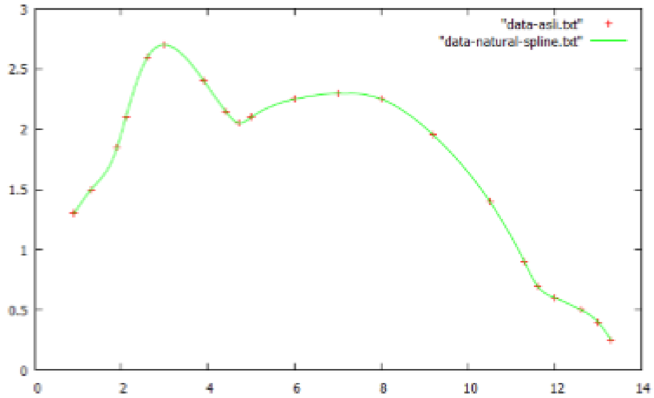
Setelah itu percobaan dilanjutkan dengan Clamped Cubic Spline, dan hasil terlihat seperti pada fig. 3 di bawah ini :

Fig. 3. Hasil perhitungan koefisien Clamped Cubic Spline

Koefisien Fungsi Interpolasi Clamped Cubic Spline

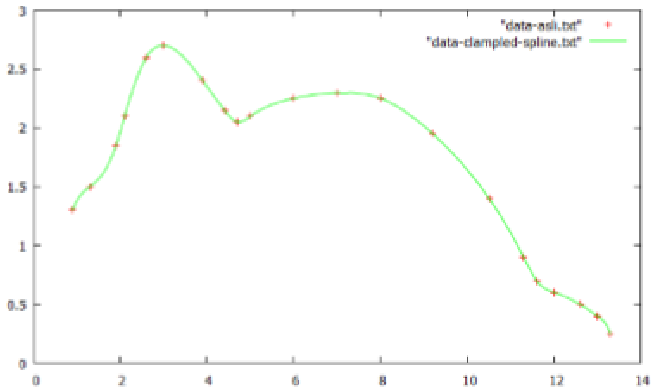
a	b	c	d
1.3	1	-1.94628	1.74069
1.5	0.27851	0.142551	0.609146
1.85	1.10745	1.23901	-2.6313
2.1	1.2873	-0.339765	-0.469666
2.6	0.595285	-1.04426	0.452632
2.7	-0.0228635	-0.501106	0.173489
2.4	-0.503277	-0.0326872	0.0784843
2.15	-0.477101	0.0850393	1.31396
2.05	-0.0713089	1.2676	-1.58117
2.1	0.262337	-0.155449	0.0431122
2.25	0.0807751	-0.0261127	-0.00466235
2.3	0.0145626	-0.0400998	-0.0244628
2.25	-0.139025	-0.113488	0.0175077
1.95	-0.335764	-0.0504604	-0.012849
1.4	-0.532105	-0.100571	-0.0194342
0.9	-0.730333	-0.147214	1.19811
0.7	-0.49517	0.93109	-0.795412
0.6	-0.132096	-0.0234043	-0.057023
0.5	-0.221766	-0.126046	0.13865
0.4	-0.25605	0.0403346	-2.845

Fig. 4. Hasil ploting koefisien Natural Cubic Spline menggunakan gnuplot



Untuk menggambarkan plot dari hasil perhitungan program Clamped Cubic Spline sama seperti pada percobaan ploting untuk Natural Cubic Spline, digunakan library tambahan dari gnuplot. Yang membedakan Clamped Cubic Spline, seperti yang telah dijelaskan di awal-awal, terdapat batasan yang harus terpenuhi. Untuk percobaan pertama, batasan yang digunakan adalah $f'(x_0) = 1$ dan $f'(x_n) = -1$. Sehingga menghasilkan plot seperti pada fig. 5

Fig. 5. Hasil ploting koefisien Clamped Cubic Spline dengan gnuplot dengan $f'(x_0) = 1$ dan $f'(x_n) = -1$



Setelah hasil ploting keluar, percobaan dengan batasan lain dapat dilakukan. Misal ketika nilai $f'(x_0) = 2$ dan $f'(x_n) = -2$, sehingga akan menghasilkan plot seperti fig. 6. Atau ketika nilai $f'(x_0) = -2$ dan $f'(x_n) = 2$, sehingga akan menghasilkan plot seperti fig. 7.

V. KESIMPULAN

Berdasarkan hasil dari percobaan diatas, dapat disimpulkan bahwa hasil perhitungan plot dengan menggunakan Natural Cubic Spline seperti yang terlihat pada fig. 4 terlihat hampir menyerupai dengan grafik aslinya. Sedangkan perhitungan plot dengan menggunakan Clamped Cubic Spline seperti yang terlihat pada fig. 5, 6, 7 terdapat grafik yang melenceng sehingga tidak menyerupai dengan grafik aslinya.

Fig. 6. Hasil ploting koefisien Clamped Cubic Spline dengan gnuplot dengan f'(x_0) = 2 dan f'(x_n) = -2

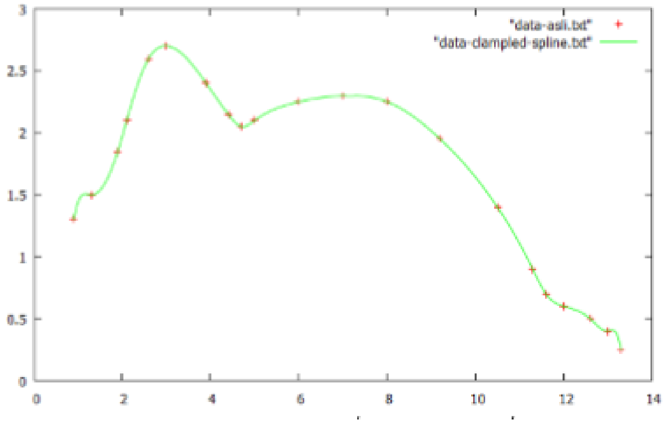
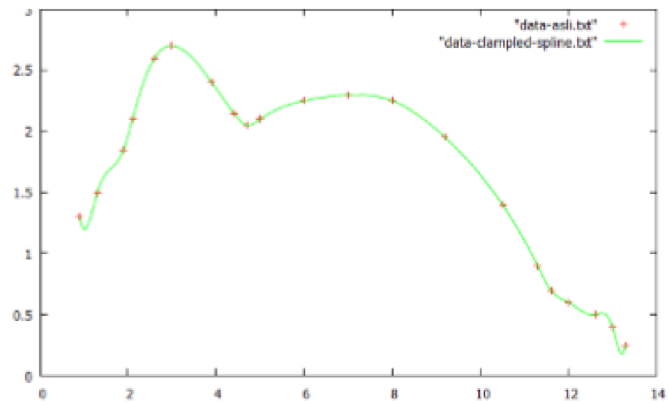


Fig. 7. Hasil ploting koefisien Clamped Cubic Spline dengan gnuplot dengan f'(x_0) = -2 dan f'(x_n) = 2



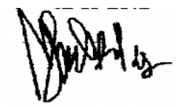
REFERENCES

[1] R.L. Burden, J. Douglas, "Numerical Analysis", 9ed, Brook/Cole Cengage Learning, Canada, 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 4 Mei 2016



Muhammad Indra N.S.
23515019