

Penyelesaian Fungsi Kontinu menggunakan Decrease and Conquer

Abdurosyid Broto Handoyo (13510107)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510107@std.stei.itb.ac.id

ABSTRAK

Penyelesaian sebuah persamaan dari fungsi tidak selalu mudah apabila menggunakan aljabar. Pendekatan numerik biasa dilakukan bila pendekatan aljabar sangat sulit untuk memecahkannya. Pendekatan numeric ternyata dapat memanfaatkan Decrease and Conquer bila diketahui selang keberadaan solusi sehingga penyelesaiannya lebih mangkus. Namun, untuk menyelesaikan sebuah fungsi dengan Decrease and Conquer fungsi tersebut harus kontinu dan monoton pada selang yang didefinisikan. Sehingga kita harus membuktikan terlebih dahulu apakah fungsi tersebut ada solusinya pada selang yang ditentukan.

Kata kunci: Decrease and Conquer, fungsi kontinu, fungsi monoton

I. PENDAHULUAN

Pada bidang matematika, seringkali kita dihadapkan dengan persoalan pencarian solusi sebuah persamaan. Persamaan matematika yang akan kita bahas dalam hal ini adalah $f(x) = C$. Dengan x adalah solusi yang akan kita cari dengan diberikan sebuah konstanta C .

Persamaan yang sederhana seperti $2x + 3 = 0$, $\sin x \cos x - 0.5 = 0$ atau $3x\sqrt{x} - 8 = 0$ dapat langsung kita tentukan persamaan untuk menentukan x . Namun, bagaimana bila $f(x)$ yang diberikan cukup sulit untuk kita buat persamaan x nya. Seperti fungsi $p * e^{-x} + q * \sin(x) + r * \cos(x) + s * \tan(x) + t * x^2 + u = 0$ sangat sulit untuk dapat kita ubah menjadi $x = g(p, q, r, s, t, u)$. Sehingga pendekatan aljabar cukup sulit diterapkan pada $f(x)$ yang rumit, tetapi optimal untuk $f(x)$ yang sederhana.

Kita membutuhkan suatu pendekatan lain untuk menyelesaikan persoalan ini secara optimal. Salah satu pendekatan lain yang dapat dilakukan adalah menggunakan pendekatan numerik dalam permasalahan ini, yaitu dengan mencoba kemungkinan x sedemikian sehingga memenuhi $f(x)$. Tetapi domain $x \in R$, membuat cara pencarian ini tidak mangkus apabila kita mencoba semua kemungkinan x yang mungkin.

Pendekatan numerik tersebut ternyata dapat kita modifikasi dengan menerapkan decrease and conquer

sehingga pencarian x dapat dilakukan secara optimal. Dengan decrease and conquer, kita juga harus menentukan terlebih dahulu batas atas dan bawah dari domain variabel x dan tingkat keakuratan dari hasil pencarian ini. Dengan menerapkan decrease and conquer, kita dapat mendapatkan kompleksitas $O(\log n)$ dengan n adalah range x yang akan kita cari.

II. DESKRIPSI MASALAH

Permasalahan yang akan dibahas kali ini adalah mencari solusi x dari sebuah persamaan $f(x) = C$ bila x berada pada sebuah selang $[a, b]$. Hasil x yang akan kita kalkulasi ini harus dapat memiliki tingkat keakuratan ($\epsilon = 10^{-9}$). Asumsi yang digunakan pada masalah ini adalah solusi pasti berada pada selang $[a, b]$

Batasan pencarian solusi x dari $f(x) = C$ yang akan kita bahas kali ini, yaitu:

1. $f(x)$ adalah fungsi kontinu
2. $f(x)$ bersifat monoton naik ataupun monoton turun pada selang a dan b
3. Domain variabel x di ketahui antara dua buah bilangan bulat a dan b

Berikut ini adalah deskripsi formal dari batasan masalah yang kita definisikan

2.1. Fungsi Kontinu

Fungsi kontinu adalah sebuah fungsi yang nilainya tidak dapat berubah mendadak. Dengan kata lain, harus memenuhi definisi kekontinuan di satu titik :

Andaikan f terdefinisi pada suatu selang terbuka mengandung c . Kita menyatakan f kontinu di c jika

$$\lim_{x \rightarrow c} f(x) = f(c)$$

Dengan definisi ini kita dapat mengambil 3 syarat:

1. $\lim_{x \rightarrow c} f(x)$ ada
2. $f(c)$ ada
3. $\lim_{x \rightarrow c} f(x) = f(c)$

Ada yang dimaksud di sini adalah nilainya terdefinisi.

Jika salah satu dari ketiga syarat tersebut tidak terpenuhi maka f diskontinu di c .

Kekontinuan pada titik ini dapat menjadi dasar untuk menentukan kekontinuan pada sebuah selang tertutup. Maka agar sebuah fungsi memenuhi kekontinuan pada suatu selang $[a, b]$ harus memenuhi definisi sebagai berikut

Fungsi f dikatakan kontinu di kanan di a jika

$$\lim_{n \rightarrow a^+} f(x) = f(a)$$
 Fungsi f dikatakan kontinu di kiri di b jika

$$\lim_{n \rightarrow b^-} f(x) = f(b)$$

Fungsi f dikatakan kontinu pada selang tertutup $[a, b]$ jika kontinu pada setiap titik pada selang (a, b) , kontinu kanan di a dan kontinu kiri di b .

Berikut ini adalah contoh fungsi-fungsi yang termasuk fungsi kontinu:

- $\sin x, \cos x$
- Fungsi polinomial
- e^x

2.2. Fungsi Monoton

Untuk membuktikan apakah sebuah fungsi monoton pada sebuah selang $[a, b]$ maka kita akan memakai teorema kemonotonan

Misalkan f kontinu pada selang I dan terdifferensiasi pada setiap titik dalam dari I , maka berlaku:

- Jika $f'(x) > 0$ untuk semua titik-titik pada selang I , maka f naik pada I
- Jika $f'(x) < 0$ untuk semua titik-titik pada selang I , maka f turun pada I

Bila $f(x)$ memenuhi syarat yang ditentukan pada 2.1 dan 2.2 maka fungsi tersebut secara valid dapat dijadikan input program.

III. PENYELESAIAN

Decrease and Conquer adalah sebuah desain algoritma dengan mereduksi persoalan menjadi sub persoalan yang lebih kecil, kemudian menyelesaikan persoalan yang lebih kecil dengan rekursif. Berbeda dengan Divide and Conquer, pada Decrease and Conquer kita tidak perlu memproses semua sub-persoalan dari reduksi persoalan yang utama, melainkan hanya 1 sub-persoalan saja.

Decrease and Conquer terdiri dari 2 tahap :

1. Decrease
Mereduksi persoalan menjadi persoalan yang lebih kecil
2. Conquer
Memproses satu sub persoalan secara rekursif

Berbeda dengan Divide and Conquer yang memiliki satu tahap tambahan, yaitu Combine. Ini disebabkan karena pada Divide and Conquer, kita harus menyelesaikan semua sub-persoalan yang dihasilkan kemudian digabung menjadi solusi permasalahan keseluruhan. Sedangkan pada Decrease and Conquer, kita hanya perlu menyelesaikan satu sub-persoalan untuk mendapatkan solusi dari permasalahan secara keseluruhan.

Untuk menyelesaikan permasalahan yang sudah dijelaskan pada Bab II, kita dapat memodelkannya sebagai permasalahan Decrease and Conquer varian decrease by constant factor. Pada persoalan kali ini, faktor pembagi yang dipilih adalah 2, yaitu setiap persoalan akan direduksi menjadi setengah dari besar persoalan yang semula.

Ide ini mirip sekali dengan binary search sebuah nilai pada array terurut. Namun, pada kasus ini kita harus mencari nilai x sehingga sedekat mungkin memenuhi $f(x) = C$ dengan tingkat kesalahan maksimal ($\epsilon = 10^{-9}$).

Untuk memudahkan pencarian, $f(x) = C$ akan diubah menjadi bentuk $g(x) = 0$ dengan $g(x) = f(x) - C$. Sehingga **kalkulasi** kita akan lebih mudah, karena untuk semua $g(x)$, kita akan membandingkannya langsung dengan nilai 0. Ini akan memudahkan proses realisasi pembuatan kode.

Sebelum kita melakukan pencarian, kita harus membuktikan fungsi yang akan kita cari memiliki solusi pada range yang kita berikan. $g(x)$ memiliki solusi pada range $[a, b]$, jika fungsi $g(x)$ kontinu, monoton dan beririsan dengan sumbu x (dengan kata lain $g(x) = 0$) pada range $[a, b]$.

Untuk membuktikan fungsi $g(x)$ beririsan dengan sumbu x pada range $[a, b]$ apabila $g(x)$ kontinu dan monoton, kita harus mencari nilai $g(a)$ dan $g(b)$. Sehingga berlaku kedua poin berikut:

- [1] Bila $g(a) < 0$ dan $g(b) > 0$, maka x memiliki solusi pada $g(x) = 0$
- [2] Bila $g(a) > 0$ dan $g(b) < 0$, maka x memiliki solusi pada $g(x) = 0$

Pernyataan [1] dan [2] adalah benar dikarenakan bila $g(x)$ adalah fungsi kontinu, maka dalam transisi dari nilai positif ke negative ataupun negative ke positif pasti akan melewati titik nol. Meski solusi dari x di titik nol mungkin saja tidak hanya satu. Sehingga kita bisa mendapatkan bentuk lebih umum dari [1] dan [2] untuk menyatakan apakah $g(x) = 0$ ada solusinya pada range $[a, b]$, yaitu:

$$g(a) \cdot g(b) < 0$$

Pada kondisi awal pencarian, kita assign batas bawah dan batas atas dari range solusi yang dicari. Range solusi pada definisi persoalan dinyatakan sebagai $[low, high]$.

low adalah batas bawah dari range solusi, sedangkan *high* adalah batas atas dari range solusi. Dalam hal ini harus berlaku $f(low) < f(high)$, sehingga $f(low) < 0$ dan $f(high) > 0$.

Dari kedua nilai *low* dan *high*, kita akan menentukan titik tengah dengan $\frac{low+high}{2}$ yang diberi nama *mid*. Dari titik *mid*, hitung nilai dari $g(mid)$. Bila $g(mid) < 0$ maka batas bawah, akan kita set pada titik tengah. Bila sebaliknya, maka titik atas akan diset pada titik tengah.

Step yang kita lakukan ini adalah step decrease karena persoalan yang awalnya [*low, high*] dapat menjadi [*low, mid*] ataupun [*mid, high*]. Step ini akan diulang terus-menerus hingga selisih antara *low* dan *high* menjadi kurang dari 10^{-9} , sehingga kita dapat yakin bahwa nilai *mid* dari *low* dan *high* tersebut, cukup akurat.

Algoritma tersebut dapat ditulis dengan pseudocode berikut:

```

WHILE (selisih Low dan High lebih dari ε)
  Mid ← nilai tengah High dan Low
  IF f(Mid) < 0
    Low ← Mid
  ELSE
    High ← Mid

```

Untuk menghitung kompleksitas dari algoritma di atas, kita akan membuat fungsi rekurensinya, yaitu sebagai berikut:

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

Pada fungsi di atas, dapat kita lihat bahwa persoalan $T(n)$ direduksi menjadi setengahnya $T(n/2)$. Sedangkan komputasi untuk mereduksi persoalan tersebut adalah 1 kali komputasi. n menyatakan range solusi yang diberikan kepada kita.

Dengan menggunakan teorema master, kita dapatkan kompleksitas dari Decrease and Conquer untuk menyelesaikan nilai x dari sebuah fungsi kontinu adalah $O(\log_2 n)$. Sehingga kita dapatkan algoritma ini mangkus untuk menyelesaikan persoalan pencarian solusi x pada $f(x) = 0$

IV. REALISASI

Dengan menggunakan bahasa C++, realisasi dari algoritma tersebut dapat dituliskan sebagai berikut:

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define E 2.7182818284
int p,q,r,s,t,u,c,a,b;
double absolute(double d)
{
  if(d<0)
    return -d;
}

```

```

return d;
}
double f(double x)
{
  return p*pow(E,-x) + q*sin(x)
    + r*cos(x) + s*tan(x) + t*x*x + u -
c;
}
int main()
{
  double hi,lo,mid;
  // meminta masukan
  cin >> a;
  cin >> b;
  cin >> p;
  cin >> q;
  cin >> r;
  cin >> s;
  cin >> t;
  cin >> u;
  cin >> c;
  //cek apakah terdapat solusi
  if(f(b)*f(a)>0) {
    printf("Tidak ada solusi x di
[%d,%d]\n", a, b);
  } else {
    hi = b;
    lo = a;
    // nilai lo harus yang negatif
    if(f(b)<f(a))
    {
      hi = a;
      lo = b;
    }
    // algoritma decrease and conquer
    while(absolute(hi-lo)>1e-9)
    {
      mid = (hi+lo)/2.00;
      if(f(mid)<0)
        lo = mid;
      else
        hi = mid;
    }
    printf("%.4lf\n", (hi+lo)/2.00);
  }
  return 0;
}

```

Pada realisasi tersebut, fasilitas fungsi yang dapat diakomodasi adalah $p.e^{-x}$, $q.\sin x$, $r.\cos x$, $s.\tan x$, $t.x^2$ dan konstanta u . Dengan p, q, r, s, t, u dan c adalah input program untuk menentukan $f(x) = 0$. Seluruh fungsi tersebut adalah fungsi kontinu. Sehingga, pertambahan fungsi kontinu juga merupakan fungsi kontinu.

V. TESTING

Program yang telah direalisasikan, akan ditest dengan beberapa testcase dengan fungsi-fungsi kontinu. Pengetesan program menggunakan 3 fungsi seperti berikut:

[1] $f(x) = 0$ dengan $f(x) = -2x^2 + 1$ pada selang $[0, 4]$

[2] $f(x) = 1$ dengan $f(x) = e^{-x} - x^2 + 3$ pada selang $[0, 1]$

[3] $f(x) = 0$ dengan $f(x) = e^{-x} - \sin x + \cos x - \tan x - x^2 + 1$ pada selang $[0, 1]$

Pada kasus [1], cek terlebih dahulu apakah $f(0)$ dan $f(4)$ memiliki tanda yang berbeda (positif dan negatif atau negatif dan positif). Karena $f(0) = -1$ dan $f(4) = 15$, maka kasus [1] pasti memiliki solusi x pada range $[0, 4]$. Kemudian kita masukkan input ini kepada program yang telah dibuat.

```
D:\Kuliah\Stima>a
Masukkan p, q, r, s, t, u dan c:
0 0 0 0 -2 1 0
Masukkan a dan b sebagai range solusi [a, b]:
0 4
Solusi x : 0.7071
```

Apabila $f(x) = 0$ kita jabarkan, maka

$$-2x^2 + 1 = 0 \leftrightarrow x^2 = 1/2 \leftrightarrow x = \sqrt{1/2} = 0.7071$$

Terbukti, untuk kasus [1], program mengeluarkan jawaban yang benar.

Pada kasus [2], ubah terlebih dahulu bentuk $f(x) = 1$ menjadi $g(x) = 0$ dengan $g(x) = f(x) - 1$, maka $g(x) = e^{-x} - x^2 + 2$. Kemudian cek terlebih dahulu apakah $g(0)$ dan $g(1)$ memiliki tanda yang berbeda. Kita dapatkan $g(0) = 3$ dan $g(1) = 1.3679$. Karena $g(0).g(1) > 0$, maka tidak ada solusi $g(x) = 0$ pada range $[0, 1]$.

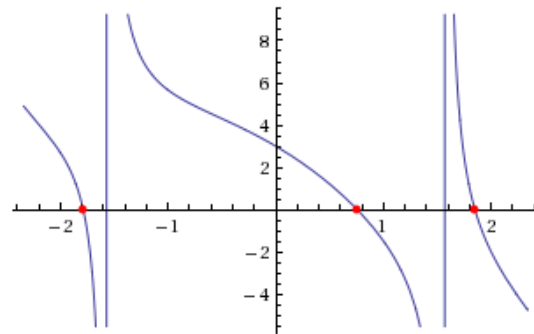
```
D:\Kuliah\Stima>a
Masukkan p, q, r, s, t, u dan c:
1 0 0 0 -1 3 1
Masukkan a dan b sebagai range solusi [a, b]:
0 1
Tidak ada solusi x di [0,1]
```

Pada kasus [3], karena $f(x) = g(x)$, maka tidak perlu mengubah $f(x)$. Kemudian cek terlebih dahulu apakah $g(0)$ dan $g(1)$ memiliki tanda yang berbeda. Kita dapatkan $g(0) = 3$ dan $g(1) = -0.7554$. Karena $g(0).g(1) < 0$, maka terdapat solusi $g(x) = 0$ pada $[0, 1]$. Kemudian jalankan program pada testcase [3]:

```
D:\Kuliah\Stima>a
Masukkan p, q, r, s, t, u dan c:
1 -1 1 -1 -1 1 0
Masukkan a dan b sebagai range solusi [a, b]:
0 1
Solusi x : 0.7554
```

Pemecahan $g(x)$ secara aljabar akan sangat rumit, maka untuk memverifikasi jawaban yang diberikan oleh program, kita akan menggunakan Wolfram Alpha. Plot gambar dan solusi fungsi yang didapat adalah seperti berikut:

Root plot:



Numerical solutions:

$$x \approx -1.78755226914969...$$

$$x \approx 0.755376684270765...$$

$$x \approx 1.84920720786915...$$

$$x \approx 4.76087098553722...$$

Pada plot dan hasil perhitungan dari WolframAlpha, didapatkan solusi $g(x) = 0$ pada $[0, 1]$ adalah 0.7554 (pembulatan). Bersesuaian dengan batasan program dan hasil program yang telah didesain.

Dengan testcase [1], [2] dan [3] berjalan dengan benar, implementasi yang dibuat dapat berjalan sesuai konsep Decrease and Conquer yang telah dijelaskan pada Bab Penyelesaian. Dengan demikian, implementasi tersebut dapat dipakai dalam menangani masalah serupa.

VI. KESIMPULAN

- Dengan menggunakan metode Decrease and Conquer, kita dapat menyelesaikan persoalan pendekatan nilai solusi dari sebuah fungsi kontinu secara akurat
- Algoritma Decrease and Conquer dapat menyelesaikan persoalan pendekatan nilai solusi x dari fungsi kontinu dengan mangkus yaitu dengan kompleksitas $O(\log_2 n)$
- Pembuktian mengenai fungsi kontinu, fungsi mooton dan ada tidaknya solusi pada selang yang telah ditentukan harus dilakukan sebelum eksekusi program

VII. UCAPAN TERIMA KASIH

Makalah ini dibuat untuk memenuhi tugas dari mata kuliah IF3051-Strategi Algoritma. Penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir sebagai dosen mata kuliah ini yang telah membimbing kami dalam menyelesaikan makalah ini.

Ucapan terima kasih juga kami berikan kepada semua pihak yang telah membantu dalam pembuatan makalah ini sehingga dapat diselesaikan.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. "Diktat Kuliah IF2251 Strategi Algoritmik". Program Studi Informatika Sekolah Teknik Elektro dan Informatika.
- [2] Cormen, et. al. 2002. "Introduction to Algorithms". MIT Press. McGraw-Hill Book Company
- [3] Rigdon, et.al. 2004. "Kalkulus". Penerbit Erlangga

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, 16 Desember 2012



Abdurrosyid Broto Handoyo (13510107)