

PENDEKATAN PARAMETER FUNGSI SINUSOIDAL MENGGUNAKAN TRANSFORMASI FOURIER DALAM OPTIMASI ALGORITMA LEAST SQUARE NONLINIER

Dedy Farhamsa¹

¹Jurusan Fisika FMIPA UNTAD, defanet@gmail.com

Abstrak

Dalam penyelesaian algoritma least square nonlinier, khususnya fungsi sinusoidal, penentuan kondisi inisial parameter sangat sensitif dalam menghasilkan model optimal, penyebabnya adalah banyaknya nilai minima yaitu antara θ hingga $\theta + 2n\pi$. Penelitian ini melakukan pendekatan kondisi inisial parameter ke nilai parameter optimal. Dasar pemikirannya adalah mencari frekuensi-frekuensi domain yang terdapat pada fungsi sinusoidal sesuai banyaknya suku yang dimilikinya. Transformasi Fourier digunakan untuk mencari frekuensi domain tersebut dan kemudian digunakan sebagai kondisi awal. Penelitian ini berhasil membuat algoritma menjadi lebih efisien dan menurunkan rasio error selama iterasi berjalan.

Kata Kunci : Transformasi Fourier, Curva Fitting NonLinier, Nilai inisial Parameter

I. Pendahuluan

Least Square adalah metode penentuan parameter fitting suatu fungsi untuk disesuaikan dengan sebaran data pengukuran sehingga memiliki error simpangan yang terkecil (SSE sum square error), tujuannya adalah membentuk sebuah model yang dapat merepresentasikan sebaran data dengan baik. Least Square nonlinier adalah algoritma iterasi penyesuaian parameter dengan tujuan untuk mereduksi SSE antara fungsi dan data pengukuran. Metode Levenberg-Marquardt Least Square pada dasarnya adalah kombinasi antara dua metode minimasi, metode *gradient descent* dan metode Gauss-Newton. Pada metode *gradient descent*, SSE direduksi dengan cara mengupdate parameter ke arah mana reduksi terbesar terjadi. Pada metode Gauss-Newton, SSE direduksi dengan asumsi bahwa fungsi least square bersifat *quadratic* local, dan mencari nilai minimum tiap *quadratic*. Metode Levenberg-Marquardt bekerja seperti metode *descent gradient* ketika parameter masih jauh dari nilai optimal dan bekerja seperti metode Gauss-Newton ketika parameter sudah mendekati nilai optimal.

Permasalahan Least Square nonlinier juga muncul ketika bekerja dengan fungsi sinusoidal. Parameter didalam fungsi ini sangat sensitif sehingga sangat sulit mendapatkan fungsi optimum tanpa informasi awal yang diperlukan sebelum menentukan nilai inisialnya. Dalam penelitian ini, metode FFT digunakan untuk mencari nilai parameter fungsi sinusoidal untuk menentukan inisial. Beberapa persamaan umum sinusoidal seperti superposisi dan fungsi sinus pangkat N juga diuji coba agar hasil penelitian ini dapat digunakan secara general dalam berbagai kasus fungsi sinusoidal.

II.1. Bentuk Non Linier Least Square

Model persamaan non-linier bergantung pada beberapa parameter yang tidak diketahui a_k , dimana $k = 0, 1, \dots, m$. Bentuk umum persamaan non-linier adalah

$$y = y(x; a_0, a_1, \dots, a_m) \text{ atau } y = y(x; \mathbf{a}) \quad (1)$$

Fungsi chi-square untuk persamaan non-linier ini adalah

$$\chi^2(\mathbf{a}) = \sum_{i=0}^n \left[\frac{y_i - y(x_i; \mathbf{a})}{\sigma_i} \right]^2 \quad (2)$$

Gradient χ^2 yang berhubungan dengan parameter a diasumsikan bernilai nol pada saat total selisih antara $y_i - y(x_i; \mathbf{a})$ minimum

$$\frac{\partial \chi^2}{\partial a_k} = -2 \sum_{i=0}^n \frac{[y_i - y(x_i; \mathbf{a})]}{\sigma_i^2} \frac{\partial y(x_i; \mathbf{a})}{\partial a_k} \quad (3)$$

dimana $k = 0, 1, \dots, m$

dengan mendiferensialkannya kembali, didapat

$$\frac{\partial^2 \chi^2}{\partial a_k \partial a_l} = -2 \sum_{i=0}^n \frac{1}{\sigma_i^2} \left[\frac{\partial y(x_i; \mathbf{a})}{\partial a_k} \frac{\partial y(x_i; \mathbf{a})}{\partial a_l} - [y_i - y(x_i; \mathbf{a})] \frac{\partial^2 y(x_i; \mathbf{a})}{\partial a_k \partial a_l} \right] \quad (4)$$

Persamaan (4) disebut Hessian untuk $\chi^2(\mathbf{a})$ yang apabila ditulis dalam bentuk matrix menjadi :

$$\nabla^2 \chi^2(\mathbf{a}) = 2\mathbf{J}_{(a)}^T \mathbf{J}_{(a)} + \mathbf{Q}_{(a)} \quad (5)$$

dimana $\mathbf{J}_{(a)}$ adalah matrix Jacobian

$$\begin{pmatrix} \frac{\partial y(x_1; \mathbf{a})}{\partial a_1} & \dots & \frac{\partial y(x_1; \mathbf{a})}{\partial a_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y(x_n; \mathbf{a})}{\partial a_1} & \dots & \frac{\partial y(x_n; \mathbf{a})}{\partial a_n} \end{pmatrix} \quad (6)$$

dan

$$\mathbf{Q}_{(a)} = 2 \sum_{i=0}^n [y_i - y(x_i; \mathbf{a})] \frac{\partial^2 y(x_i; \mathbf{a})}{\partial a_k \partial a_l}$$

Dalam metode Gauss-Newton untuk menyelesaikan permasalahan model non-linier, suku $\mathbf{Q}(a)$ dapat diabaikan karena menganggap $\chi^2(a)$ bernilai kecil untuk mengestimasi nilai optimal dari parameter a . Namun pendekatan ini menjadi tidak akurat apabila nilai $\chi^2(a)$ besar.

II.2. Algoritma Levenberg Marquadt

Metode Least Square untuk fungsi nonlinier merupakan fungsi simpangan terkecil kuadrat (*chi-square*) yaitu

$$\chi^2(\mathbf{a}) = \sum_{i=0}^n \left[\frac{y_i - y(x_i; \mathbf{a})}{\sigma_i} \right]^2 \quad (6)$$

Yang mana suatu fungsi nonlinier dikatakan baik jika $\chi^2(\mathbf{a}) \approx 0$. Persamaan (4) jika ditulis dalam matrix berbentuk

$$\mathbf{J}_{(\mathbf{a})}^T \mathbf{J}_{(\mathbf{a})} \Delta \mathbf{a} = -\mathbf{J}_{(\mathbf{a})}^T \mathbf{F}_{(\mathbf{a})} \tag{7}$$

dimana $\mathbf{J}(\mathbf{a})$ adalah matrix Jacobian

$$\begin{pmatrix} \frac{\partial y(x_1; \mathbf{a})}{\partial a_1} & \dots & \frac{\partial y(x_1; \mathbf{a})}{\partial a_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y(x_n; \mathbf{a})}{\partial a_1} & \dots & \frac{\partial y(x_n; \mathbf{a})}{\partial a_n} \end{pmatrix}$$

dan

$$\mathbf{F}_{(\mathbf{a})} = \begin{bmatrix} y_1 - y(x_1; \mathbf{a}) \\ \sigma_1 \\ \vdots \\ y_n - y(x_n; \mathbf{a}) \\ \sigma_i \end{bmatrix}$$

Untuk menghindari matrix yang singular, Marquardt menempatkan parameter positif λ pada algoritmanya.

$$(\mathbf{J}_{(\mathbf{a})}^T \mathbf{J}_{(\mathbf{a})} + \lambda \mathbf{I}) \Delta \mathbf{a} = -\mathbf{J}_{(\mathbf{a})}^T \mathbf{F}_{(\mathbf{a})} \tag{8}$$

Strategi algoritma dalam menyelesaikan persamaan (6) sambil menjaga kekonvergenan matrix adalah memulai memasang suatu nilai awal untuk λ dan kemudian diperkecil (misalnya dibagi dengan kelipatan 2) untuk setiap iterasi. Jika langkah ini mereduksi nilai χ^2 maka iterasi diteruskan. Namun apabila langkah ini tidak mereduksi χ^2 , perbesar nilai λ (misalnya dikali dengan kelipatan 2) untuk setiap iterasi berikutnya. Proses iterasi ini diteruskan sampai ditemukan nilai χ^2 yang terendah.

III. Transformasi Fourier

Proses secara fisik dapat digambarkan dalam domain waktu dan domain frekuensi. Domain waktu digambarkan oleh nilai h sebagai fungsi dari waktu t yaitu $h(t)$, sedangkan domain frekuensi digambarkan oleh nilai amplitudo (atau fasa untuk kasus kompleks) sebagai fungsi frekuensi f yaitu $H(f)$ dimana nilai f berkisar $-\infty < f < \infty$. Transformasi Fourier untuk kedua domain tersebut diberikan oleh

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt$$

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} dt \tag{9}$$

Biasanya t diukur dalam detik dan f dalam putaran per detik atau Herz. Jika h adalah fungsi posisi x (dalam meter), maka H merupakan fungsi inversi panjang gelombang (putaran per meret), dan seterusnya.

Untuk data N sampel (yaitu data yang dicatat selama suatu interval waktu). Persamaan (II.21) dirubah dalam bentuk diskrit

$$H_n \equiv \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \quad (10)$$

Jumlah data hasil H_n sama banyaknya dengan h_k , dari persamaan (II.22) dapat dilihat bahwa nilai dari H_1 adalah total dari data h_k . Dalam interval dua pengukuran yang sama, ada nilai frekuensi f_c yang disebut *Nyquist critical frequency*, yang diberikan oleh

$$f_c \equiv \frac{1}{2\Delta} \quad (11)$$

atau sama dengan satu putaran dalam interval dua pengukuran yang sama.

Inversi dari persamaan (II.22) diberikan oleh

$$h_k \equiv \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi i k n / N} \quad (12)$$

Jika h_k adalah real, maka persamaan diatas dapat ditulis sebagai jumlah fungsi sinus dan cosinus dengan koefisien real [6] :

$$h_k \equiv \frac{1}{N} \sum_{n=0}^{N-1} \left[a_n \cos\left(\frac{2\pi i k n}{N}\right) + b_n \sin\left(\frac{2\pi i k n}{N}\right) \right] \quad (13)$$

dimana

$$\begin{aligned} a_n &= \text{real}[H_n] \\ b_n &= -\text{imag}[H_n] \\ 0 &\leq k \leq N - 1 \end{aligned}$$

IV. Hasil dan Pembahasan

Terdapat banyak aplikasi fungsi periodik dalam sistim fisik yang sering digunakan antara lain fungsi getaran, gelombang, superposisi, interferensi dan sebagainya.

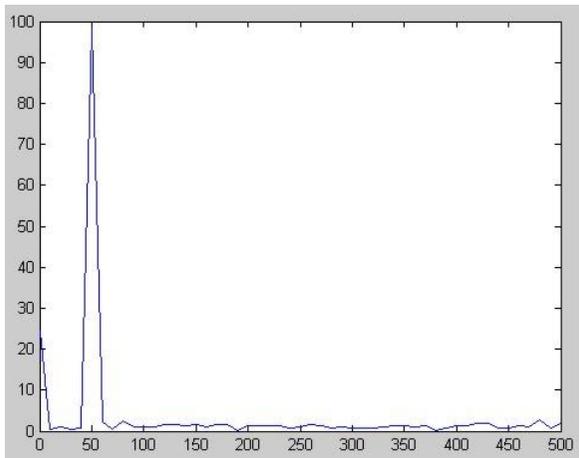
Dalam sistem getaran, fungsi yang digunakan adalah

$$f_t = A \sin(\omega t) \quad (14)$$

dimana A amplitudo dan ω kecepatan sudut dan t variable waktu. Metode LM akan mencari nilai A dan ω paling sesuai dengan trend data yang diberikan. Uji coba fungsi yang diberikan adalah $f_t = 2 \sin 341t$ dengan penyisipan nilai random sebagai noise sistem. Perhitungan frekuensi dengan metode FFT menggunakan program Matlab seperti table 1:

Tabel 1. FFT.m

```
t = 0:.0005:.099;
y = 2*sin(2*pi*50*t)+;
I = abs(fft(y))
N = length(I)
nyquist = (N/2)
f = 1000/N*(0:nyquist)
I = I(1:nyquist+1)
plot(f,I)
```



Gambar 1. Grafik FFT untuk menentukan frekuensi dominan

Grafik antara intensitas vs frekuensi menunjukkan frekuensi maksimum pada $f = 0,5\text{Hz}$ sehingga nilai inisial parameter fungsi (14) dapat ditentukan $A = 0$ dan $\omega = 2\pi f = 3,14$. Nilai inisial A dibiarkan bernilai nol karena algoritma LM tidak mempunyai masalah dalam mencari nilai parameter bebas seperti ini. Kode Matlab untuk Algoritma LM dapat dilihat pada table 2.

Tabel 2. LM.m

```
function [x,histout,costdata] = levmar(x0,f,tol,maxit)
maxarm=10;
if nargin < 4
maxit=100;
end
itc=1; xc=x0;
[fc,gc,jac]=feval(f,xc);
nu0=.001d0;
mvar=length(gc); nvar=length(xc);
nfun=1; ngrad=1; numh=0;
numf=1; numg=1; numh=0;
ithist(1,1)=norm(gc); ithist(1,2) = fc; ithist(1,4)=itc-1; ithist(1,3)=0;
ithist(itc,:);
nu=norm(gc);
while(norm(gc) > tol & itc <= maxit)
itc=itc+1;
hc=(jac'*jac)+ nu*eye(nvar);
dc=hc\gc;
xt=xc-dc;
```

```

[ xp, nup, idid ] = trtestlm( f, xc, xt, fc, jac, gc, nu, nu0 );
if idid > 20
    error('too many iterations in TR solve')
end
xc = xp; nu = nup; numf = numf + idid;
if idid > 1
    [ fc, gc, jac ] = feval( f, xc ); numf = numf + 1;
    numg = numg + 1;
end
ithist( itc, 1 ) = norm( gc ); ithist( itc, 2 ) = fc;
ithist( itc, 4 ) = itc - 1; ithist( itc, 3 ) = idid;
ithist( itc, : )
end
x = xc; histout = ithist( 1 : itc, : );
costdata = [ numf, numg, numh ];

function [ xp, nup, idid ] = trtestlm( f, xc, xt, fc, jac, gc, nu, nu0 );
mu0 = .1; mulow = .25; muhigh = .75; mup = 2.0; mdown = .5;
z = xc; nvar = length( z ); numft = 0; numgt = 0; itr = 0;
while z == xc & itr <= 20
    ft = feval( f, xt );
    itr = itr + 1;
    s = xt - xc; ared = fc - ft; pred = -gc * s;
    rat = ared / pred;
    if rat < mu0
        nu = max( nu * mup, nu0 );
        hc = ( jac * jac ) + nu * eye( nvar );
        dc = hc \ gc;
        xt = xc - dc;
    elseif rat < mulow
        z = xt; nu = max( nu * mup, nu0 );
    else
        z = xt;
        if rat > muhigh
            nu = mdown * nu;
            if nu < nu0 nu = 0.0; end
        end
    end
end
nup = nu; xp = z; idid = itr;

```

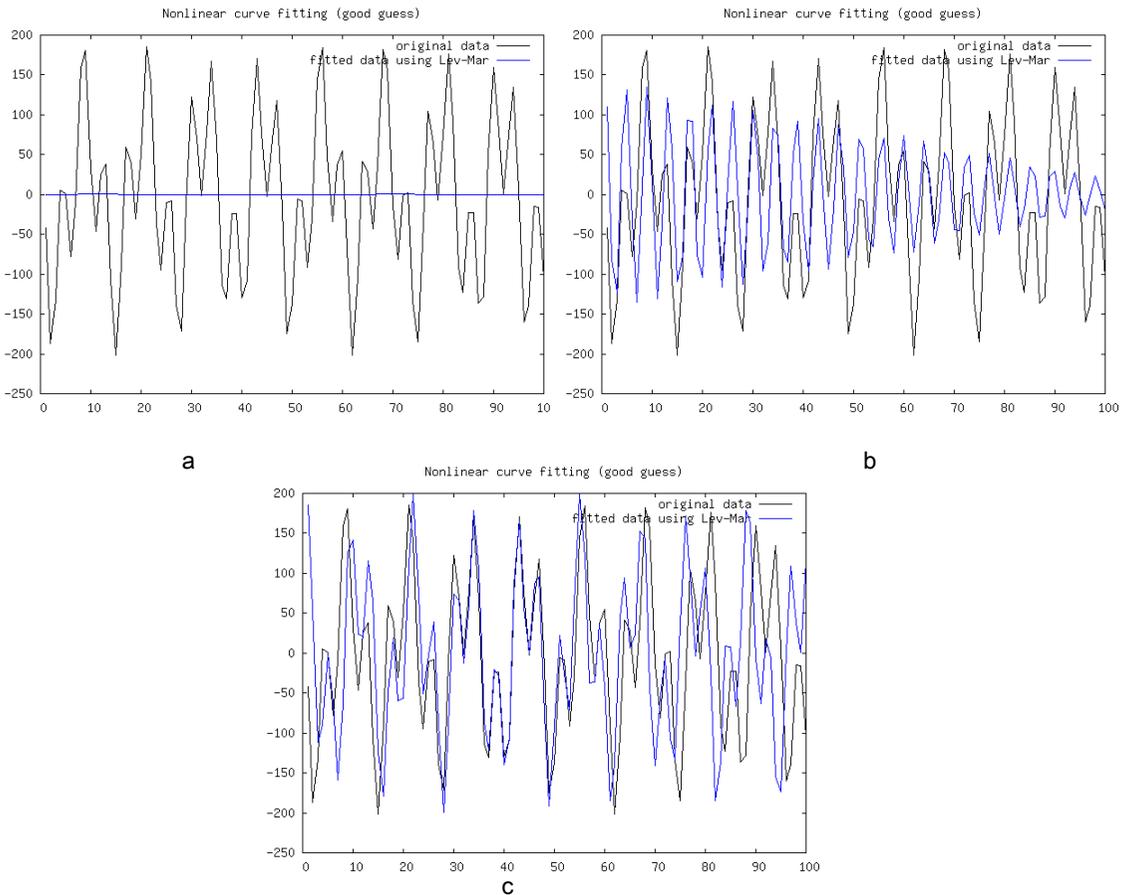
Dalam kasus superposisi fungsi sinusoidal

$$A_1 \sin(x_1 t) + A_2 \cos(x_2 t) \quad (15)$$

FFT dapat menentukan nilai inisial x_1 dan x_2 dengan tepat setelah mengambil dua frekuensi yang memiliki intensitas maksimum dan dikonversi menjadi frekuensi sudut ($x_1 = 2\pi f_1$ dan $x_2 = 2\pi f_2$) sementara nilai $A_1 = A_2 = 0$ dibiarkan bernilai nol. Kasus ini juga berlaku untuk superposisi dengan N fungsi sinusoidal, penentuan nilai inisial parameter diambil dari N nilai frekuensi sudut ($2\pi f_n$) dominan.

Contoh kasus dibawah ini mencoba untuk memodelkan fungsi $y = a \cos(bx) + b \sin(ax)$ menggunakan algoritma LM. Tiga grafik (Gambar 1,2, dan 3) menunjukkan proses penyesuaian nilai inisial awal a dan b . Gambar 1. nilai inisial $a=0$ dan $b=0$ menghasilkan curva fitting yang sangat buruk, gambar 2 diberikan inisial $a=80$ dan $b=52$ menghasilkan curva fitting yang lebih baik, hanya gambar 3 dimana inisila $a=100$ dan $b=101$ yang mendekati nilai sebenarnya memberikan curva

fitting yang sesuai. Fungsi sinusoidal memiliki nilai inisial yang sangat sensitive dalam algoritma LM sehingga proses pemilihan nilainya dilakukan secara coba-coba yang membutuhkan pengalaman dan pengetahuan khusus. Namun proses coba-coba ini dapat dihindari dengan mengambil frekuensi-frekuensi dominan yang dihasilkan oleh FFT.



Gambar 2. a, b dan c Grafik hasil fitting dengan percobaan inisial. Hanya grafik c yang mempunyai inisial mendekati parameter sebenarnya dapat menghasilkan fitting yang baik.

Dalam kasus fungsi sinusoidal berpangkat genap, misalnya :

$$A \sin^{2N}(xt) \tag{15}$$

Kerapatan frekuensinya menjadi dua kali lebih besar, sehingga untuk menentukan nilai inisial parameter menjadi setengah kali dari frekuensi sudutnya ($x = \pi f$)

V. Kesimpulan

Transformasi Fourier terbukti mampu digunakan untuk mencari nilai inisial parameter fungsi sinusoidal sehingga dapat digunakan untuk optimalisasi algoritma Levenberg-Macuard. Pada fungsi

sinus atau cosines, inisial parameter diambil dari 2 *frekuensi dominan hasil konversi FFT. Pada fungsi superposisi nilai inisial parameter N suku sinusoidal diambil dari N buah frekuensi dominan pertama dikalikan dengan fasa 2 . Sementara fungsi sinus berpangkat genap $2n$. nilai inisial diambil dari setengah frekuensi dominan (πf). Walaupun nilai inisial yang dihasilkan oleh FFT ini kemudian masih disesuaikan dalam proses algoritma LM, namun metodel ini menghasilkan nilai yang sangat dekat dengan parameter sebenarnya sehingga iterasi yang dibutuhkan dalam proses optimalisasi dapat ditekan seminimal mungkin dan waktu proses running komputer dapat berjalan lebih cepat.

VI. Daftar Pustaka

- [1]. _____, 1992, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press.
- [2]. Aster R.C, Borchers B, Clifford T.H, 2005, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press.
- [3]. Davis.C.J, 1986, *Statistics and Data Analysis in Geologi 2nd*, John Willey & Son. Inc, Canada
- [4]. Howison, Sam; Crighton (Deceased), C. G.; Ablowitz, M. J.; Davis, S. H.; Hinch, E. J.; Iserles, A.; Ockendon, J.; Olver, P. J., 2005, *Practical Applied Mathematics: Modelling, Analysis, Approximation*, Cambridge University Press.
- [5]. Jacobson, 1993, *Object Oriented Software Engineering*, Addison Wesley, USA.
- [6]. MathWork Group, 2007, *Matlab^R 7th Documentation- Data Analisis*, Mathwork, Inc.