BAB III

IMPLEMENTASI METODE MARQUARDT DALAM PENYELESAIAN MASALAH OPTIMASI PEMROGRAMAN NONLINEAR

3.1 FUNGSI NONLINEAR

Metode Marquardt sering diaplikasikan dalam menyelesaikan masalah fungsi nonlinear multivariabel. Adapun langkah - langkah penyelesaiannya mengacu pada konsep jumlahan kuadrat terkecil. Sebelum masuk pada pokok bahasan, terlebih dahulu kita tinjau fungsi nonlinear itu sendiri.

Fungsi nonlinear multivariabel yang paling sederhana adalah fungsi berderajad dua berpusat pada titik asal dan disebut bentuk kuadratik. Dan untuk fungsi multivariabel umum yang lain berderajad sebarang. Biasanya disajikan sebagai

$Y = F(\mathbf{X})$ 52

dimana $\mathbf{X} = (x_1, x_2, x_3, \ldots, x_N)^T$

adalah vektor variabel yang membangun berdimensi N

F = fungsi obyektif

Dalam banyak aplikasi praktis X harus diseleksi dari himpunan diskrit. Disini kita asumsikan F dan derivatifnya ada dan kontinu dimana - mana.

3.2 DISKRIPSI METODE MARQUARDT

3.2.1 FUNGSI NONLINEAR KUADRAT TERKECIL

Misal $f_1, f_2, f_3, ,\dots, f_M$ adalah M fungsi berharga real dari N variabel real tak diketahui. Diberikan pasangan titik-titik data yang didefinisikan oleh (p_{ui}, y_i) ,u = 1,2,...,k, i = 1,2,3,...,M. Ambil $X \in R^N$, dan kita ingin mencocokkan data tersebut dengan bentuk F(X;P) sedemikian sehingga kita dapat mengaproksimasi harga X^* yang mana akan meminimalkan :

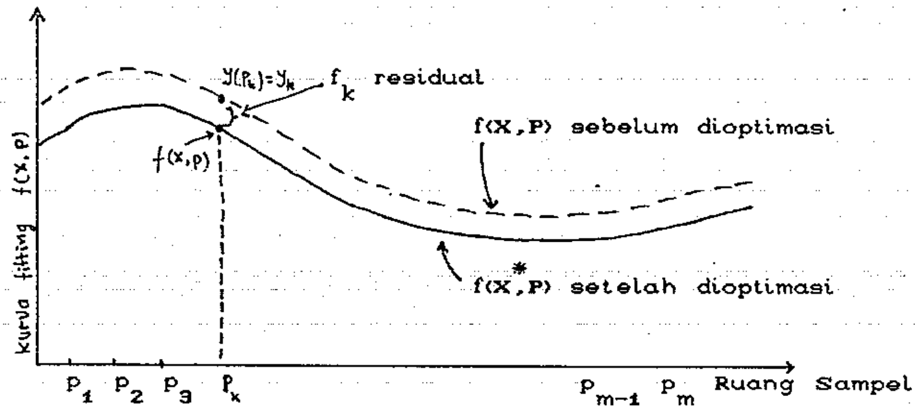
$$\phi(X) = \frac{1}{2}\sum_{i=1}^{M} (f(X;P_i) - y_i)^2 = \frac{1}{2}\sum_{i=1}^{M} f_i^2(X) \quad \text{atau}$$

$$\phi(X) = \frac{1}{2} \| F(X;P) - Y \|_2^2 = \frac{1}{2} \| F(X) \|_2^2 \qquad 53$$

adalah minimum, dimana $P_i = (p_{1i}, p_{2i}, \dots ,p_{ki})^T$. Jika kita definisikan residual yang merupakan kesalahan pada setiap titik sampel sebagai

$$F(X) = F(X;P) - Y \qquad 54$$

dan diperlihatkan dalam gambar sebagai berikut :



Gambar 2. Kurva Fitting dari aproksimasi fungsi nonlinear f(X,p)

Sekarang, relatif minimum dicari sepanjang derivatif parsial pertama dari ϕ yang merupakan gradien dari fungsi obyektif kuadrat terkecil terhadap masing-masing komponen x_j sehingga didapatkan :

$$\nabla \phi(X) = 0 \qquad 55$$

$$G_j(X) = \nabla_j \phi(X) = \frac{1}{2}\, 2 \sum_{i=1}^{M} f_i(X)\ (\nabla_j f_i(X)) \qquad 56$$

untuk $j = 1,2,3,\ldots,N$

atau $$G(X)_{N\times 1} = J^T(X)_{N\times M} F(X)_{M\times 1} \qquad 57$$

dengan definisi matriks jakobian J adalah

$$J_{ij} = \nabla_j f_i = \partial f_i(X)/\partial x_j$$

$$J_{ij} = \begin{Bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & & & \\ \frac{\partial f_i}{\partial x_1} & \frac{\partial f_i}{\partial x_2} & \cdots & \frac{\partial f_i}{\partial x_N} \\ \vdots & & & \\ \frac{\partial f_M}{\partial x_1} & \frac{\partial f_M}{\partial x_2} & \cdots & \frac{\partial f_M}{\partial x_N} \end{Bmatrix}_{M\times N}$$

dimana $j = 1,2,3,\ldots,N$ dan $i = 1,2,3,\ldots,M$

Suatu aproksimasi dibuat untuk menyederhanakan persamaan (55). Kita pikirkan ekspansi Taylor dari $\phi(X)$ di sekitar X adalah

$$\phi(X_k + \delta_t) = \phi(X_k) + G(X_k)^T \delta_t + \frac{1}{2}\delta_t^T H(X_k)\ \delta_t + O(\delta_t^3) \qquad 58$$

dimana :

k = diasosiasikan sebagai indeks suatu iterasi

t = indeks yang menyatakan hasil dari deret Taylor.

$\delta_t \quad = X - X_k$ adalah perubahan dalam X

$X_k \quad$ = adalah titik ekspansi dalam R^N

$G(X_k) = \nabla \phi(X_k)$ = vektor kolom N - komponen dari derivatif parsial pertama $\phi(X)$ yang dievaluasi pada X_k.

$H(X_k) = \nabla^2\phi(X_k)$ = matriks simetri N x N dari derivatif parsial kedua $\phi(X)$ yang dievaluasi pada X_k, sering disebut matriks Hessian. Elemen dalam baris ke - l dan kolom ke-j adalah

$$\frac{\partial^2\phi}{\partial x_l\ \partial x_j}\ ,\ \text{untuk}\ \ l,j = 1,2,3,\ldots N$$

$O(\delta_t^{\,3})$ = syarat semua dari orde tinggi

$$\frac{\nabla^2\phi}{\partial x_l \partial x_j} = \begin{bmatrix} \frac{\partial^2\phi}{\partial x_1 \partial x_1} & \frac{\partial^2\phi}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2\phi}{\partial x_1 \partial x_N} \\ \frac{\partial^2\phi}{\partial x_2 \partial x_1} & \frac{\partial^2\phi}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2\phi}{\partial x_1 \partial x_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2\phi}{\partial x_N \partial x_1} & \frac{\partial^2\phi}{\partial x_N \partial x_2} & \cdots & \frac{\partial^2\phi}{\partial x_N \partial x_N} \end{bmatrix}_{N\times N}$$

Kita ambil model linearisasi dari penderetan Taylor dari fungsi non linear tersebut , sehingga bila didifferensialkan terhadap δ_t didapatkan :

$$\nabla\phi(X_k + \delta_t) = 0 \qquad 59$$

$$\simeq G(X_k) + H(X_k)\delta_t = 0 \qquad 60$$

$$H(X_k)\delta_t = -G(X_k)$$

$$\delta_t = -H(X_k)^{-1}G(X_k)$$

$$= -H(X_k)^{-1}J(X_k)^T F(X_k) \qquad 61$$

Jika $H(X_k)$ adalah matriks non singular, persamaan (60) dapat diselesaikan untuk vektor koreksi δ_t dan minimum dapat dicapai dengan

$$X^* = X_k + \delta_t \qquad 62$$

Sehingga iterasi numerik memberikan :

$$X_{k+1} = X_k - H(X_k)^{-1}\, G(X_k) \qquad 63$$

Barisan X_1, X_2, X_3, ..., X_k, ... konvergen ke X^* yang meminimalkan $\phi(X)$, dengan harga awal X_o. Kondisi ini berlaku hanya jika dipenuhi jika $H(X_k)$ atau $J(X_k)$ non singular. Jika tidak dipenuhi maka iterasi dapat divergen sehingga kerap kali gagal mencapai konvergensi.

Untuk mengatasi itu dipergunakan vektor koreksi dan dalam praktek biasanya X dikoreksi hanya beberapa persen dari δ; karena nilai δ yang besar dapat mengakibatkan ekstrapolasi jatuh pada daerah diluar yakni dimana fungsi ϕ tidak lagi dapat direpresentasikan oleh persamaan (58) sehingga akhirnya proses mencapai divergensi.

3.2.2 APROKSIMASI HESSIAN DEFINIT POSITIF

Matriks Hessian H adalah disusun dari semua derivatif parsial kedua dari ϕ. Dengan menggunakan operator vektor del ∇ untuk differensial sebarang, ekspresi valid untuk Hessian adalah :

$$H = \nabla\,(\nabla\,\phi)^T = \nabla^2\phi \qquad 64$$

Notasi $\nabla^2\phi$ adalah simbol untuk matriks dari differensial pasial orde dua dari ϕ. Jadi dari persamaan (64) dan (56) menghasilkan

$$H = \nabla \left[\sum_{i=1}^{M} f_i (\nabla f_i) \right]^T = \nabla^2 \phi$$

$$H = \sum_{i=1}^{M} \left[(\nabla f_i)(\nabla f_i)^T + f_i (\nabla^2 f_i) \right] \qquad 65$$

hasil ini equivalen dengan

$$H = J^T J + M \qquad 66$$

dimana matriks $M_{N\times N}$ berisi residual dan derivatif keduanya yaitu $M = \sum_{i=1}^{M} f_i \nabla^2 f_i$ 67

dimana $f_i = f(X;P_i) - y_i$

Matriks M ini sering diperbincangkan bahwa persamaan (66) dapat diaproksimasi oleh hanya syarat suku pertamanya; matriks J^TJ definit positif, karena suku kedua (M) mendekati nol jika residual f_i dekat dengan nol. Residual akan dekat dengan nol pada saat aproksimasi dekat dengan solusi.

Tetapi, secara umum $\sum_{i=1}^{M} f_i \nabla^2 f_i$ tidak diabaikan dan sehingga prosedur iterasi numerik harus digunakan untuk mencapai perbaikan dengan aproksimasi.

Permulaan memanggil langkah Newton dalam penyelidikan untuk solusi X^* yang meminimalkan fungsi obyektif $\phi(X)$:

$$H(X)\,\delta = -G(X)$$

$$\nabla^2 \phi(X)\,\delta = -G(X)$$

$$(J^TJ + M)\,\delta = -J^TF \qquad 68$$

dengan mengambil $M \longrightarrow 0$ maka didapat metode Newton yaitu

$$J^TJ\,\delta = -J^T F \qquad 69$$

Metode Newton tersebut bekerja sangat baik bila model nonlinearnya tidak rumit atau perkiraan awal dekat dengan minimum. Kontur konstanta ϕ dalam ruang variabel dari model linear adalah berbentuk ellips (gbr.3). Untuk model nonlinear, kontur permukaan ϕ sangat terdistorsi (gbr.4) yakni tidak lagi mendekati bentuk ellips maka pencapaian konvergensi memakan waktu yang sangat lama. Tetapi dalam hampiran minimum ϕ, konturnya sangat dekat dengan ellips.

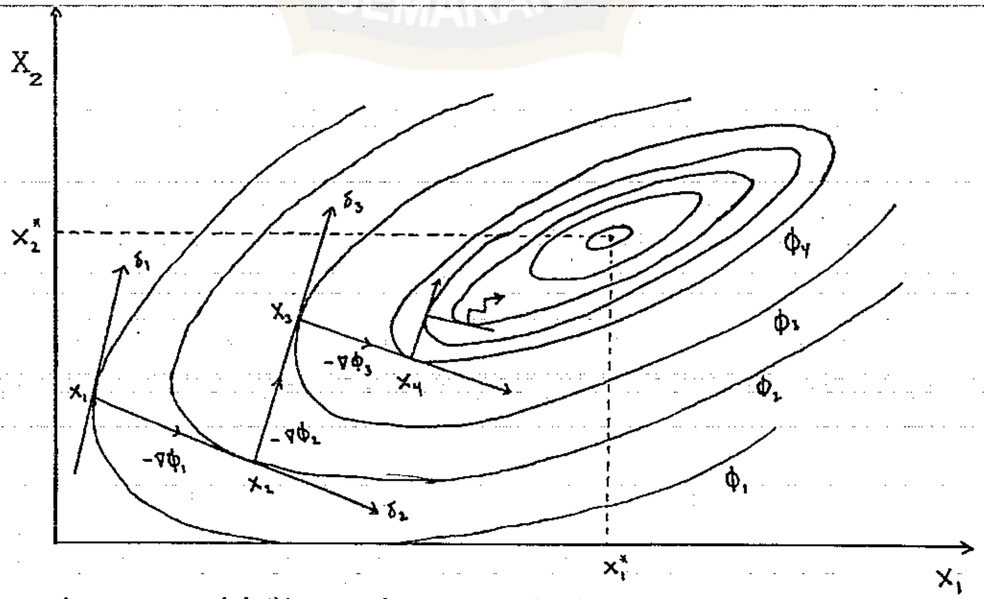
Sebagaimana biasanya, metode Steepest Descent (Gradien) beranjak dari titik perkiraan dalam arah negatif dari gradien ϕ. Jadi

$$\delta_g = -\nabla\phi \qquad 70$$

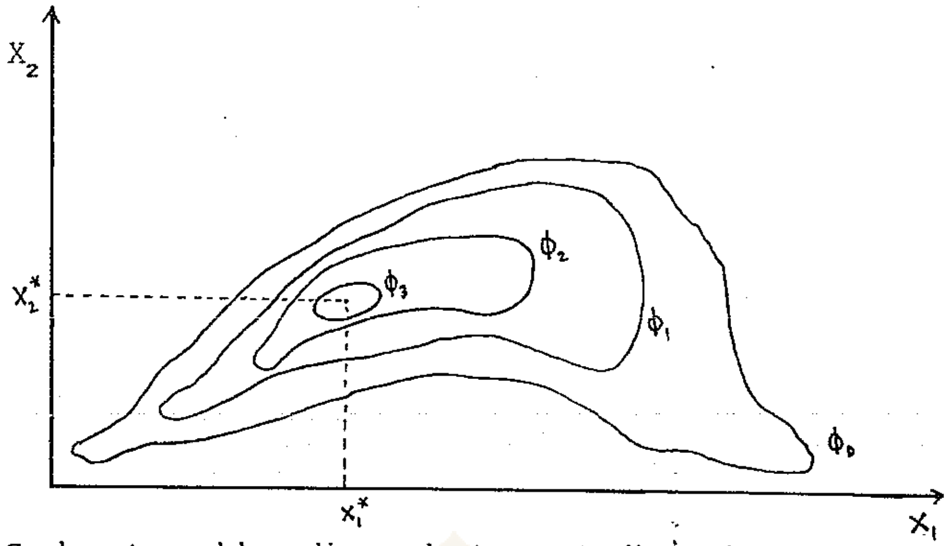
$$\delta_g = -\left[\frac{\partial\phi}{\partial x_1}, \frac{\partial\phi}{\partial x_2}, \ldots, \frac{\partial\phi}{\partial x_N}\right]^T \qquad 71$$

$$= -J(X)^T F(X) \qquad 72$$

dimana δ_g = menyatakan vektor arah dari penyelidikan gradien.



Gambar 3. Model linear, konturnya berbentuk ellips



Gambar 4. model nonlinear, konturnya terdistorsi

Metode Newton dalam persamaan (60) atau (69) diatas adalah ekuivalen dengan metode Steepest Descent dibawah bentuk ellips :

$$\| \delta \|_2 = (\delta^T H \delta)^{1/2} \qquad 73$$

Dengan cara lain, di bawah bentuk ellips metode Steepest Descent dicapai ketika $H = I$, matriks identitas. Secara geometri intepretasi semacam tadi diberikan H^{-1} menyimpangkan vektor δ dari vektor normal untuk hiperplane. Jadi titik terletak dimana bidang adalah tangen untuk ellips satuan.

Pertimbangan yang diperoleh dari metode Steepest Descent dan Metode Newton adalah bahwa metode yang baik harus menghasilkan vektor koreksi yang arahnya sekitar 90^o dari gradien negatif dari ϕ (lihat gambar 3). Jika tidak maka nilai ϕ pada titik-titik sepanjang vektor koreksi akan membesar; sebenarnya diharapkan yang bernilai kecil.

Pertimbangan lain adalah bahwa pada masalah yang umum, perpanjangan dari permukaan ϕ sangat terdistorsi sehingga δ_t (vektor koreksi hasil perhitungan metode Taylor) dan δ_g (vektor metode Gradien) biasanya hampir saling tegak lurus. Menurut metode Marquardt, sudut antara δ_t dan δ_g bernilai antara $80^o < \gamma < 90^o$. Dari perhitungan diatas dapat kita yakini bahwa dalam metode yang lebih baik dapat diperoleh hubungan antara δ_t dan δ_g .Juga jelas bahwa dalam metode gradien dan metode Newton pemilihan arah vektor koreksi dilakukan sebelum penentuan ukuran langkah (step size). Nanti nampak dalam algoritma Marquardt, bahwa arah vektor koreksi dan ukuran langkah dapat ditentukan secara simultan.

Maka Marquardt memberikan metode untuk equivalensi tersebut, dan dari ekspresi Hessian dalam persamaan (66) membawa persamaan penting untuk formulasi Marquardt. Algoritma Marquardt menjamin kekonvergenan, pada harga penambahan $\lambda \geq 0$ yang kecil, dengan mengijinkan kemungkinan penambahan berkali -kali pada matriks satuan untuk $H(X_k)$. Jadi iterasi pada persamaan (62) menjadi

$$X_{k+1} = X_k - \{(H(X_k) + \lambda_k I)\}^{-1} J(X_k)^T F(X_k) \qquad 74$$

3.2.3 KONSEP DASAR METODE MARQUARDT

Dalam algoritma Marquardt F(X) fungsi real diuraikan dalam deret Taylor dengan mengabaikan suku berorde tinggi sehingga diperoleh :

$$\langle F(X_k + \delta)\rangle = F(X_k) + \nabla F(X_k)\delta + \frac{1}{2}\delta^T \nabla^2 F(X_k)\delta + O(\delta^3) \qquad 75$$

dimana :

$\delta \quad = \quad X - X_k$ adalah perubahan dalam X

$X_k \quad =$ adalah titik ekspansi dalam R^N

$J(X_k) = \nabla F(X)$ = matriks berorde $M \times N$ dari derivatif parsial pertama $F(X)$ yang dievaluasi pada X_k.

$\nabla^2 F(X)$ = matriks simetri NxN dari derivatif parsial $F(X)$ yang dievaluasi pada X_k ,sering disebut matriks Hessian. Elemen dalam baris ke - ι dan kolom ke-j adalah

$$\frac{\partial^2 F}{\partial x_\iota \ \partial x_j} = J^T J.$$

$O(\delta^3) \quad =$ syarat semua dari orde tinggi

Sedangkan tanda kurung siku digunakan untuk membedakan prediksi yang dibuat berdasarkan atas model yang telah dilinearkan. Maka nilai F didekati oleh ruas kanan persamaan (75).

Kita linearisasi dengan penderetan Taylor dari fungsi nonlinear tersebut , dan memberikan $X \in R^N$ yang akan akan meminimalkan

$$\psi(\delta) = \| \langle F(X_k + \delta) \rangle \|_2 \qquad 76$$

sebagai fungsi dari δ , maka $X_k + \delta$ diuraikan sebagai solusi.

ψ adalah fungsi nonlinear dalam δ , sekarang kita linearisasi $F(X_k + \delta)$ dan dicapai masalah kuadrat terkecil linear

$$\psi(\delta) = \| F(X_k) + \nabla F(X_k)\, \delta \|_2 \qquad 77$$

dari penyelidikan, linearisasi tidaklah valid untuk semua harga δ , dan kita pikirkan masalah kuadrat terkecil linear berkendala :

$$\min \{ \psi (\delta) : \| D \delta \|_2 \le \varepsilon \} \qquad 78$$

Dalam teori D adalah suatu matriks nonsingular , tetapi dalam implementasi D adalah matriks diagonal yang memperhatikan penskalaan dari permasalahannya. Dilain pihak , δ dalam hyper ellips

$$\Omega = \{ \delta : \| D \delta \|_2 \le \varepsilon \}, \qquad 79$$

tetapi jika D adalah diagonal, maka Ω memiliki sumbu arah sepanjang koordinat dan panjang sumbu semi ke-j adalah ε/d_j. Dengan d_j elemen ke-j dari matriks diagonal D.

Kita pikirkan sekarang solusi dari persamaan (78), secara umum, dan jadi masalahnya adalah

$$\min \{ \| F + J \delta \|_2 : \| D \delta \|_2 \le \varepsilon \} \qquad 80$$

dimana $F \in R^M$ dan J adalah matriks $M \times N$. Dasar untuk metode Marquardt adalah bahwa jika δ^* merupakan solusi dari persamaan (80) maka $\delta^* = \delta(\lambda)$ untuk suatu $\lambda \ge 0$ dimana

$$\delta (\lambda) = - (J^T J + \lambda D^T D)^{-1} J^T F. \qquad 81$$

Pandang persamaan (77) berbentuk linear dalam δ, karena itu X^* dapat ditemukan dengan metode kuadrat terkecil standar. Minimum, X^*, diperoleh dari

$$\nabla \phi (X) = J(X)^T F(X) = 0 \qquad 82$$

Ambil bentuk kuadratik dari persamaan (58), yaitu:

$$\emptyset_k(\delta) = \phi(X_k) + G(X_k)^T \delta + \frac{1}{2} \delta^T H (X_k) \delta \qquad 83$$

dimana $\emptyset_k(\delta)$ adalah hassilpemotongan deret Taylor dari $\emptyset(X)$ disekitar X_k , dimana $\delta = X - X_k$.

Kita pikirkan masalah konvensional yang mana :

$$\Omega_k = \{ X \;:\; \| X - X_k \|_2 \le \varepsilon_k \}, \qquad 84$$

dan δ_k merupakan penyelesaian dari sub masalah, yaitu :

$$\underset{\delta}{\text{Min}}\ \emptyset_k(\delta) \text{ berlaku untuk } \| D\,\delta \|_2 \le \varepsilon_k \qquad 85$$

Masalah berkendala dalam persamaan (85) direlasikan dengan fungsi langrange sebagai berikut :

$$L(\delta,\lambda) = \emptyset_k(\delta) + \tfrac{1}{2}\lambda((D\delta)^T.D\delta - \varepsilon_k^2)$$

$$L(\delta,\lambda) = \phi(X_k) + G(X_k)^T\,\delta + \frac{1}{2}\delta^T\, H(X_k)\,\delta$$
$$+\tfrac{1}{2}\lambda((D\delta)^T.D\delta - \varepsilon_k^2) \qquad 86$$

Syarat perlu orde pertama untuk minimal lokal adalah titik stasioner dari fungsi lagrange, dengan mengambil gradien $\nabla L(\delta,\lambda) = 0$, menghasilkan

$$G(X_k) + H(X_k)\,\delta + \lambda\, D^2\,\delta = 0$$

$$(H(X_k) + \lambda\, D^2)\,\delta = -\,G(X_k)$$

$$\delta = -\,(H(X_k) + \lambda\, D^2)^{-1} G(X_k) \qquad 87$$

dimana $\delta = \delta_k$ dan $\lambda = \lambda_k \ge 0$ memberikan titik stasioner. Selama δ_k merepakan minimal lokal maka $\nabla L(\delta_k,\lambda_k) = 0$ dan

$$\emptyset_k(\delta) \ge \emptyset_k(\delta_k).$$

Selanjutnya pandang deret Taylor dari $\emptyset$ disekitar δ_k:

$$\emptyset(\delta_k + tS) = L(\delta_k + tS,\lambda_k)$$
$$= L(\delta_k,\lambda_k) + t\nabla L(\delta_k,\lambda_k)S + t^2 S\nabla^2 L(\delta_k,\lambda_k)S + 0(S^3) \qquad 88$$

dimana $\nabla L(\delta_k,\lambda_k) = 0$

$$\nabla^2 L(\delta,\lambda) = H(X_k) + \lambda D^2$$

$$\nabla^2 L(\delta_k,\lambda_k) = H(X_k) + \lambda_k D^2$$

Sehingga δ_k minimal lokal maka

$$L(\delta_k + tS,\lambda_k) \ge L(\delta_k,\lambda_k) \qquad 89$$

untuk t cukup kecil. Jadi

$$\lim_{t \to 0} \frac{L(\delta_k + tS, \lambda_k) - L(\delta_k, \lambda_k)}{t^2} = \frac{1}{2} S^T (H(X_k) + \lambda_k D^2) S \geq 0$$

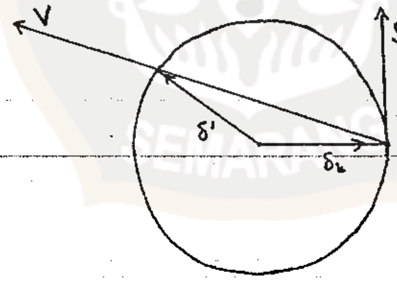
didapat $S^T(H(X_k) + \lambda_k D^2)S \geq 0,\ \forall\ S,\ S^T\delta_k = 0$ 90

Ini merupakan syarat perlu orde dua untuk solusi lokal. Persamaan (90) terlihat bahwa vektor S dan δ_k ortogonal yaitu $S^T\delta_k = 0$ atau

$$\text{Cos}\ \gamma_k = \frac{S^T\ \delta_k}{\|\ S\ \|_2 \|\ \delta_k\|_2} = 0, \text{ sehingga } \gamma_k = 90^o \qquad 91$$

Selanjutnya diambil δ_k sebagai solusi global persamaan (85) dan V suatu vektor sedemikian sehingga $V^T\delta_k \neq 0$, maka dapat dibangun vektor $\delta' \neq \delta_k$ yang memenuhi persamaan (85), dengan

$$\delta' - \delta_k \propto V \longrightarrow \delta' \propto \delta_k + V \qquad 92$$



Gambar 4 Membangun Vektor δ'

Berdasarkan asumssi solusi global maka

$$\emptyset_k(\delta_k) \leq \emptyset_k(\delta') = L(\delta', \lambda_k) \qquad 93$$

Dari persamaan (88) dan (92) memenuhi bahwa

$$L(\delta_k+V,\lambda_k) = L(\delta_k,\lambda_k) + \nabla L(\delta_k,\lambda_k)V + \frac{1}{2}V^T\nabla^2 L(\delta_k,\lambda_k)V + 0(V^3) \qquad 94$$

$$L(\delta_k+V,\lambda_k) = L(\delta_k,\lambda_k) + \frac{1}{2}V^T\nabla^2 L(\delta_k,\lambda_k)V + 0(V^3)$$

mencapai minimum ketika

$$L(\delta_k+V,\lambda_k) \geq L(X_k,\lambda_k), \quad \text{yang berarti}$$

$$V^T(H(X_k) + \lambda_k D^2)\, V \geq 0 \quad \forall\, V : V^T\delta_k \neq 0$$

Hubungan persamaan (90) dan (94), terpenuhi syarat perlu order dua untuk solusi global adalah bahwa $H(X_k)+ \lambda_k D^2$ adalah semi definit positif. Selanjutnya pandang persamaan (81) dipenuhi pada δ_k untuk $\lambda = \lambda_k$ dan ambil $H(X_k)+ \lambda_k D^2$ adalah definit positif. Maka untuk suatu $\delta \neq \delta_k$ yang fisibel dalam persamaan (85) memenenuhi persamaan (88) yaitu :

$$\emptyset_k(\delta) = L(\delta,\lambda_k) > \emptyset_k(\delta_k) \qquad 95$$

dimana $H_k+ \lambda_k D^2$ definit positif.

Jadi syarat cukup order dua untuk solusi global tunggal adalah bahwa persamaan (81) terpenuhi pada δ_k untuk suatu $\lambda = \lambda_k$ dan $H_k + \lambda_k D^2$ adalah definit positif.

Teorema 20

Jika untuk λ ($\lambda \geq 0$) sebarang dan δ_o memenuhi persamaan :

$$(H + \lambda I)\, \delta_o = -G \qquad 96$$

maka δ_o akan meminimumkan $< \phi >$ pada bola berjejari $\| \delta \|_2$ yang memenuhi

$$\| \delta \|_2^2 = \| \delta_o \|_2^2 \qquad 97$$

Bukti :

Untuk menemukan δ yang akan meminimumkan :

$$< \phi(\delta) > = \| <F> - Y \|_2^2 \qquad 98$$

$$= \| F_o + J\,\delta - Y \|_2^2$$

dan memenuhi

$$\| \delta \|_2^2 = \| \delta_o \|_2^2 \qquad 99$$

dimana $\quad Y \quad$: fungsi yang diobservasi

$\langle\phi\rangle$: aproksimasi minimum dari ϕ

F_o : fungi yang dikalkulasi dengan harga δ_o

Syarat perlu untuk titik stasioner dari fungsi pengali lagrange, yaitu :

$$\frac{\partial L}{\partial \delta_1} = \frac{\partial L}{\partial \delta_2} = \frac{\partial L}{\partial \delta_3} = \ldots = \frac{\partial L}{\partial \delta_i} = 0 \;, \quad \frac{\partial L}{\partial \lambda} = 0 \qquad 100$$

dalam hal ini λ adalah pengali lagrange.

$$L(\delta,\lambda) = \| F_o + J\delta - Y \|_2^2 + \lambda \left(\| \delta \|_2^2 - \| \delta_o \|_2^2 \right) \qquad 101$$

dengan syarat perlu diperoleh :

$$\frac{\partial L(\delta,\lambda)}{\partial \delta} = 2J^T(F_o + J\delta - Y) + 2\lambda\delta = 0 \qquad 102$$

$$\Leftrightarrow \quad J^T(F_o - Y) + J^TJ\delta + \lambda\delta = 0$$

$$\Leftrightarrow \quad (J^TJ + \lambda I)\delta = -J^T(F_o - Y) \qquad 103$$

dan $$\frac{\partial u(\delta,\lambda)}{\partial \lambda} = \| \delta \|_2^2 - \| \delta_o \|_2^2 = 0 \qquad 104$$

Jadi persamaan (96) dan (103) merupakan persamaan yang identik, karena sudah jelas dari uraian sebelumnya bahwa $(J^TJ + \lambda I)$ definit positif.

Pemikiran lebih lanjut melingkupi penskalaan secara implisit dari variabel dalam ruang **X**. Sifat - sifat δ_t dari $H\,\delta_t = -G$ invarian terhadap transformasi linear dalam ruang **X** . Sedang sifat dari Steepest Descent tidak invarian terhadap skala pengukuran, selama $H = I$ tidak dapat ditransformasi linear dalam ruang variabel **X**. Dan juga modifikasi metode Newton biasanya tidak invarian, sebab $H + \lambda I$ tidak dapat ditransformasi linear dengan tepat ketika $\lambda > 0$. Sehingga perlu penskalaan yang tepat di ruang dalam ruang X, yaitu X' = DX. Untuk ruang X diskala dalam satuan standart deviasi dari turunan $\partial f_i/\partial x_j$, untuk sejumlah titik sampel i = 1,2,3,...,M. Karena turunannya tergantung pada X_j, dan nilai X_j dicoba-cobakan dipakai untuk mrnghitung turunannya. Pandang

$$D_k = \text{diag}\ (\ d_{1k}\ d_{2k}\ d_{3k}\ \ldots\ d_{Nk}\ )$$

dimana

$$d_{jk} = \|\ \partial_j F(X_o)\|_2\ ,\ k \geq 0 \qquad 105$$

Pemilihan ini biasanya memadai sepanjang $\|\ \partial\ F(X_k)\ \|_2$ tidak membesar dengan k. Tetapi jika $\|\ \partial\ F(X_k)\ \|_2$ membesar, maka dikehendaki menurun sepanjang ε/d_j dari sumbu semi ke - j dari hiperellips (79), ketika F menaik dengan terlalu cepat sepanjang variabel ke-j. Oleh karena itu, langkah komponen ke -j yang besar cenderung tak dapat dipercayai.

Alasan ini menghantarkan untuk memilih

$$d_{jo} = \|\ \partial_j F(X_o)\ \|_2 \qquad 106$$

$$d_{jk} = \max\ \{\ d_{j(k-1)},\ \|\partial_j F(X_k)\|_2\}\ ,\ k \geq 1.$$

Catatan bahwa $\|\ \partial_j F(X_k)\ \|_2$ menurun hanya memperlihatkan bahwa F tidak berubah demikian cepat sepanjang variabel ke-j. Selanjutnya tidak dikehendaki menurun dalam d_j

$$d_{jk} = \|\ \partial_j F(X_k)\ \|_2\ ,\ k \geq 0 \qquad 107$$

secara perhitungan ini lebih rendah dari persamaan (105) dan (106). Catatan bahwa persamaan (105),(106) dan (107) membuat algoritma Marquardt skala invarian.

Pemilihan ini menyebabkan untuk menskala matriks H, yang ditransformasikan kedalam matriks yang mempunyai koefisien - koefisien korelasi sepanjang $\partial f_i / \partial X_j$. Selanjutnya didefinisikan matriks terskala H' , dan vektor terskala G' :

$$H' = (h'_{ij}) = \left[\frac{h_{jj'}}{\sqrt{h_{ij}}\ \sqrt{h_{j'j'}}} \right]$$

$$G' = (g_j) = \left[\frac{g_{jj'}}{\sqrt{h_{ij}}} \right]$$

diselesaikan untuk koreksi deret Taylor menggunakan persamaan :

$$H' \ \delta_t = - G'$$

maka $\delta_j = \delta'_j / \sqrt{h_{ij}}$

Bentuk persamaan pada iterasi ke - k :

$$(H'_k + \lambda_k I) \ \delta'_k = - G' \qquad 108$$

Transformasi linear dari jarak variabel adalah

$$X' = D \ X \quad \text{dan} \quad \delta'_o = D \ \delta_o \qquad 109$$

Pandang $J = [\nabla_j f_k]$, $k = 1,2,\ldots,M$, $j = 1,2,\ldots,N$

dengan dalil rantai jakobian dari ruang variabel baru X' adalah

$$\frac{\partial f_k}{\partial X'_j} = \frac{\partial f_k}{\partial X_j} \cdot \frac{\partial X_j}{\partial X'_j} \ , \ \text{dimana} \ \frac{\partial X_j}{\partial X'_j} = d_{jj}^{-1}$$

didapat matriks jakobian dalam ruang variabel baru X' adalah $J' = J \ D^{-1}$

dalam ruang variabel baru, pikirkan δ' didefinisikan sebagai

$$[(J')^T J' + \lambda \ I \ ] \ \delta'_o = - (J')^T F$$

$$[(JD^{-1})^T \ JD^{-1} + \lambda I \ ]^{-1} \ D\delta_o = - (J \ D^{-1})^T \ F$$

$$[J^T (D^{-1})^T D^{-1} J + \lambda I] D\delta_o = - D^{-1} J^T F \qquad 110$$

mereduksi menjadi

$$(J^T J + \lambda D^2)\delta_o = - J^T F \qquad 111$$

Theorema 21

Jika $\delta(\lambda)$ adalah solusi dari persamaan (96) untuk suatu nilai λ maka $\| \delta(\lambda) \|_2^2$ adalah fungsi turun kontinu dari λ sedemikian sehingga $\lambda \to \infty$, $\| \delta(\lambda) \|_2^2 \to 0$.

Bukti :

H matriks simetris definit positif, maka dapat ditransformasikan dengan sumbu rotasi ortogonal, menjadi matriks diagonal $D = diag(d_1, d_2, \ldots, d_N)$ tanpa mengubah jarak antara titik-titiknya. $(H + \lambda I)$ adalah matriks simetris, dimana $S_1, S_2, \ldots, S_N$ merupakan vektor eigennya yang orthogonal, yaitu $S_i^T S_j = 0$, $i \neq j$. S_i juga orthonormal maka $S^T S = I$. Jika $S^T H S = D$, dimana $S^T S = I$ dan semua elemen diagonalnya $d_1 \geq d_2 \geq \ldots \geq d_N$ adalah positif. Misalkan $d_1, d_2, \ldots, d_N$ adalah nilai eigen dari H, berarti

$$H\, S_j = d_j S_j \;, \; j = 1,2,\ldots,N \qquad 112$$

diselesaikan dengan $Det\,(H - DI) = 0$,

$$Det \begin{bmatrix} h_{11} - d & h_{12} & \ldots & h_{1N} \\ h_{21} & h_{22} - d & \ldots & h_{2N} \\ \vdots & & & \\ h_{N1} & h_{N2} & \ldots & h_{NN} - d \end{bmatrix} = 0$$

dihasilkan polinomial derajad N , yaitu

$$(d - d_1)(d - d_2)(d - d_3) \ldots (d - d_N) = 0$$

sehingga diperoleh d_1, $d_2, \ldots, d_N$. Pendekatan langkah Marquardt δ dalam persamaan (96) menggunakan dekomposisi harga singular, dengan pemisahan matriks jakobian

$$J = U\ D^{1/2}\ S^T$$

dimana U dan S adalah matriks diagonal

$D^{1/2}$ meliputi harga singular dari J pada diagonal. selama $U^TU = I$, $S^TS = I$,

$$J^TJ = (U\ D^{1/2}\ S^T)U\ D^{1/2}\ S^T = S^T\ D^{1/2}\ U^TU\ D^{1/2}\ S^T$$

$$= SDS^T = H$$

sehingga $H + \lambda I = S(D + \lambda I)S^T = S\ W\ S^T$

matriks W adalah diagonal dan mempunyai invers, maka

$$\delta_o = -\ S\ (\ W\ )^{-1}\ S^T\ G \qquad 113$$

$$\delta_o = -\ S\ (\ D + \lambda\ I\ )^{-1}\ S^T\ G \qquad 114$$

dengan mendefinisikan $V = S^TG$, maka :

$$\|\ \delta_o(\lambda)\ \|_2^2 = G^TS\ (D + \lambda I)^{-1}S^TS\ (\ D + \lambda I)^{-1}\ S^TG$$

$$= V^T\ [\ (\ D + \lambda\ I\ )^2]^{-1}\ V \qquad 115$$

$$= V^T\ (W^2)^{-1}V$$

menggunakan dekomposisi Spektral dapat ditulis sebagai

$$\|\ \delta_o(\lambda)\ \|_2^2 = \sum_{j=1}^{N} \frac{V_j^2}{W_j^2}$$

$$= \sum_{j=1}^{N} \frac{V_j^2}{(\ d_j + \lambda\ )^2} \qquad 116$$

Persamaan tersebut merupakan fungsi turun dari λ ($\lambda \geq 0$) sedemikian sehingga $\lambda \longrightarrow \infty$, $\|\ \delta_o(\lambda)\ \|_2^2 \longrightarrow 0$. ∎

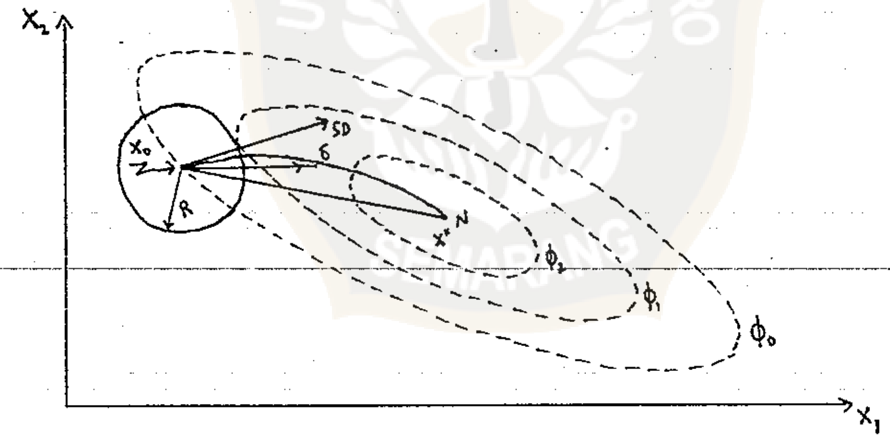
Ternyata fungsi $\delta(\lambda)$ adalah rasional dengan pole (singular) pada $\lambda = -\ d_j$, $j = 1,2,\ldots,N$. Persamaan (111) merepresentasikan penskalaan secara implisit dari variabel. Hal ini berguna dalam hubungannya dengan interpretasi daerah terbatasi sebagai prediksi δ pada model kuadratis sedemikian sehingga $\|\ \delta\ \|_2^2 \leq R$, R adalah konstanta yang bergantung pada pemilihan skala ruang X. Elemen-elemen $D = [d_{ij}]$ adalah harga akar kuadrat rata-rata dari derivatif pertama residual yaitu :

$$d_{ij} = \left[(\nabla_j f_1)^2 + (\nabla_j f_2)^2 + \ldots + (\nabla_j f_N)^2 \right]^{1/2}$$

atau

$$D^2 = J^T J \qquad 117$$

Persekitaran kepercayaan berpusat dengan radius R, digunakan untuk menghalangi zig zag yang tak dapat dipercayai dalam arah penyelidikan Steepest Descent, sebelum sampai pada persekitaran dari minimum, maka kecepatan konvergensi dari metode Newton untuk mengatasinya.



Gambar 5. Kurva trayektori hubungan titik Newton dan titik Steepest Descent

Jadi langkah Newton didekati dengan panjang R, dengan kecepatan beralasan maju, dan supaya penyelidikan menuruni lembah yang diinginkan dengan mengalikan panjang langkah dengan faktor konstanta kecil. Adapun kebijaksanaan daerah terbatasi disini adalah :

1. Jika titik Steepest Descent diluar persekitaran, maka langkah jarak R dalam arah Stepest Descent.
2. Jika titik Steepest Descent didalam persekitaran, maka langkah untuk titik Newton jika titik tersebut didalam persekitaran atau jarak R dalam arah itu.
3. Jika pemilihan langkah gagal mereduksi harga fungsi, selanjutnya kalikan panjang langkah dengan 4^{-1} sampai langkah menghasilkan fungsi mereduksi.

Dalam penyelidikan, permukaan nonlinear diaproksimasi secara kuadratis sekitar minimal lokal. Radius kepercayaan R dipilih secara bijaksana dengan indikator yang mengijinkan penggunaan pengaproksimasian dari radius kepercayaan.

Adapun indikator yang digunakan adalah :

i. Perbandingan antara reduksi aktual dan reduksi prediksi yang dicapai pada setiap langkah, sebagai perbaikan ε.

reduksi aktual : $\Delta\phi(\delta_k) = \|F(X_k)\|_2^2 - \|F(X_k + \delta_k)\|_2^2$

reduksi prediksi : $\Delta\emptyset(\delta_k) = \|F(X_k)\|_2^2 - \|F(X_k)+F'(X_k+\delta_k)\|_2^2$

maka rasio diberikan sebagai

$$R(\delta_k) = \Delta\,\phi(\delta_k)/\Delta\,\emptyset(\delta_k) \qquad 118$$

Rasio ini menjamin keakuratan $\emptyset(X_k)$ dalam mengaproksimasi $\phi(X_k+\delta_k)$.

Pandang persamaan (111)

$$(J^TJ + \lambda D^2)\delta_o = -J^TF$$

$$J^TJ\delta_o + \lambda D^2\delta_o = -J^TF$$

$$2J^TJ\delta_o^2 + 2\lambda D^2\delta_o^2 = -2J^TF\,\delta_o$$

$$\|J\delta_o\|_2^2 + 2\lambda\|D\delta_o\|_2^2 = \|J\delta_o\|_2^2 - 2J^TF\,\delta_o + \|F\|_2^2 - \|F\|_2^2$$

$$= \| F \|_2^2 - \| F + J\delta_o \|_2^2$$

Sehingga $R(\delta_k)$ dapat ditulis

$$R(\delta_k) = \frac{\| F(X_k) \|_2^2 - \| F(X_k + \delta_k) \|_2^2}{\| J\delta_k \|_2^2 + 2\lambda \| D\delta_k \|_2^2}$$

$$R(\delta_k) = \frac{1 - \left\| \dfrac{F(X_k + \delta_k)}{F(X_k)} \right\|_2^2}{\left\| \dfrac{J\delta_k}{F(X_k)} \right\|^2 + 2\left(\dfrac{\lambda^{1/2} \| D\delta_k \|_2}{\| F(X_k) \|_2} \right)^2} \qquad 119$$

hal ini mengakibatkan

$$\| J\delta_k \|_2 \le \| F(X_k) \|_2, \quad \lambda^{1/2} \| D\delta_k \|_2 \le \| F(X_k) \|_2$$

Penyebut tak menghasilkan overflow, dan penyebut tak peduli non negatif dari kesalahan roundoff. Sedangkan pembilang boleh overflow jika

$$\| F(X_k + \delta_k) \|_2^2 \ge \| F(X_k) \|_2^2 \text{ maka } R(\delta_k) \le 0.$$

ii. Penggunaan sudut antar arah penyelidikan negatif gradien dan arah penyelidikan Newton. berdasarkan persamaan (29) diberikan

$$\operatorname{Cos} \gamma = \frac{-\delta^T G}{\| \delta \|_2 \| G \|_2} \qquad 120$$

dimana δ langkah Newton dalam persamaan (61) dalam praktek, akan definit menuju $80^o < \gamma < 90^o$, yang menghasilkan trayektori dekat dengan tangen kurva (tegak dengan gradien). Dibawah kondisi γ harga fungsi munurun menuju terminasi penyelidikan.

Lebih jelasnya diberikan hubungan δ_o , δ_t, dan δ_g

yang dibahas dalam teorema berikut :

Teorema 22

Jika γ sudut antara δ_o dan δ_g maka γ ialah fungsi turun monoton dalam λ sedemikian sehingga $\lambda \to \infty$, $\gamma \to 0$. Karena δ_g tidak bergantung pada λ, maka δ_o berotasi terhadap δ_g sementara $\lambda \to \infty$.

Bukti :

Tinjau persamaan :

$$\delta_g = \left[\sum_{i=1}^{M} (y_i - f_i) \frac{\partial f_i}{\partial x_j} \right]^T, \quad j = 1,2,\ldots,N \qquad 121$$

sehingga

$$\delta_g = - G \qquad 122$$

Dari definisi

$$\cos \gamma = \frac{-\delta^T G}{(\| \delta \|_2) (\| G \|_2)} \qquad 123$$

$$= \frac{\{S (D + \lambda I)^{-1} S^T G \}^T G}{(V^T ((D + \lambda I)^2)^{-1} V)^{1/2} . (G^T G)^{1/2}}$$

$$= \frac{\{S (D + \lambda I)^{-1} V \}^T G}{(V^T ((D + \lambda I)^2)^{-1} V)^{1/2} . (G^T G)^{1/2}}$$

$$= \frac{V^T (D + \lambda I)^{-1} S^T G}{(V^T ((D + \lambda I)^2)^{-1} V)^{1/2} . (G^T G)^{1/2}}$$

$$= \frac{V^T (D + \lambda I)^{-1} V}{(V^T ((D + \lambda I)^2)^{-1} V)^{1/2} . (G^T G)^{1/2}} \qquad 124$$

$$= \frac{\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)}}{\left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2} \right]^{1/2} (G^T G)^{1/2}}$$

Dengan mendiferensialkan dan menyederhanakan akan diperoleh :

$$\frac{\partial \cos\gamma}{\partial \lambda} = \frac{\left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{d_j+\lambda}\right]\left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j+\lambda)^3}\right] - \left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2}\right]^2}{\left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2}\right]^{3/2} \left(G^T G \right)^{1/2}}$$

dikalikan dengan $\dfrac{\left[\prod_{j=1}^{N} (d_j + \lambda)^2\right]^2}{\left[\prod_{j=1}^{N} (d_j + \lambda)^2\right]^2}$ maka didapat

$$\frac{\partial \operatorname{Cos} \gamma}{\partial \lambda} = \frac{\xi - \chi}{\left[\displaystyle\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2}\right]^{3/2} \left(G^T G \right)^{1/2} \left[\displaystyle\prod_{j=1}^{N} (d_j + \lambda)^2\right]^2}$$

$$\xi = \left[\sum_{j=1}^{N} \frac{V_j^2}{d_j+\lambda} \prod_{j=1}^{N} (d_j + \lambda)\right]\left[\sum_{j=1}^{N} \frac{V_j^2}{(d_j+\lambda)^3} \prod_{j=1}^{N} (d_j+\lambda)^3\right]$$

$$\chi = \left[\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2} \prod_{j=1}^{N} (d_j + \lambda)^2\right]^2$$

$$\frac{\partial \cos \gamma}{\partial \lambda} = \frac{\left[\sum_{j=1}^{N} V_j^2 \, \Pi_{1j} \right]\left[\sum_{j=1}^{N} V_j^2 \, \Pi_{3j} \right] - \left[\sum_{j=1}^{N} V_j^2 \, \Pi_{2j} \right]^2}{\left[\sum_{j=1}^{N} \frac{V_j^2}{(d_j + \lambda)^2} \right]^{3/2} \left[\prod_{j=1}^{N} (d_j + \lambda)^2 \right]^2 \left(G^T G \right)^{1/2}} \qquad 125$$

dimana

$$\Pi_{1j} = \prod_{\substack{j'=1 \\ j' \neq j}}^{N} (d_{j'} + \lambda) \qquad \Pi_{2j} = \prod_{\substack{j'=1 \\ j' \neq j}}^{N} (d_{j'} + \lambda)^2 ,$$

$$\Pi_{3j} = \prod_{\substack{j'=1 \\ j' \neq j}}^{N} (d_{j'} + \lambda)^3$$

Penyebut dalam persamaan (125) bernilai positif, karena masing - masing faktor bernilai positif. Oleh karena itu tanda dari $d \operatorname{Cos}\gamma / d\lambda$ adalah tanda dari pembilang. Dengan catatan bahwa

$$\Pi_{1j}\Pi_{3j} = (\Pi_{2j})^2$$

dan pertidaksamaan Schwartz's adalah

$$\| U \|_2^2 \; \| V \|_2^2 - (U^T V)^2 = \| U \|_2^2 \; \| V \|_2^2 - \| U \|_2^2 \; \| V \|_2^2 \cos^2\alpha$$

$$= \| U \|_2^2 \; \| V \|_2^2 \; (1 - \cos^2\alpha) \geq 0 \qquad 126$$

dimana α adalah sudut antara **U** dan **V** maka

$$W = \left[\sum_{j=1}^{N} \left(V_j \Pi_{1j}^{1/2} \right)^2 \right] \left[\sum_{j=1}^{N} \left(V_j \Pi_{3j}^{1/2} \right)^2 \right] - \left[\sum_{j=1}^{N} \left(V_j \Pi_{1j}^{1/2} \right) \left(V_j \Pi_{3j}^{1/2} \right) \right]^2$$

didapat $W \geq 0$, sehingga

$$\frac{\partial \cos \gamma}{\partial \lambda} = \frac{W}{\left[\sum_{j=1}^{N} \frac{V_j^{\,2}}{(d_j \;+\lambda)^2}\right]^{3/2} \left[\prod_{j=1}^{N} (d_j+\lambda)^2\right]^2 \left(G^T \; G \right)^{1/2}} \qquad 127$$

Dengan pertidaksamaan Schwartz's di atas, persamaan (126) adalah positif. Jadi $d \cos \gamma / d \lambda$ adalah selalu positif ($\lambda > 0$). Akibatnya γ merupakan fungsi turun monoton dari λ.

Untuk nilai λ yang besar , matriks $(H + \lambda I)$ didominasi oleh diagonal λI. Jadi nampak dari persamaan $(H + \lambda I)\delta_o = - G$ bahwa jika $\lambda \dashrightarrow \infty$, $\delta_o \dashrightarrow -G/\lambda$, dalam hal ini δ_o dan G menjadi pendekatan yang proforsional, sedemikian sehingga sudut antaranya mendekati nol. Pada sisi yang lain Jika $\lambda = 0$ dalam persamaan :

$$(H + \lambda \; I \quad)\delta_o = - G \; ,$$

maka vektor δ_o dan G bertemu pada sudut tertentu yaitu : $0 \leq \gamma \leq \pi/2$. Itu menunjukkan bahwa γ merupakan fungsi continu monoton turun dari λ, sedemikian sehingga jika λ ----> ∞ maka γ --> 0.

3.2.4 KONTROL PARAMETER MARQUARDT

Dalam implementasi $\lambda>0$ merupakan parameter Marquardt, sebagai parameter arah yang digunakan untuk mengoptimasi minimum sepanjang trayektori kurva.

Pandang

$$\phi(X_k+t_k\delta_k) = \phi(X_k) + G(X_k)^T t_k\delta_k + \frac{1}{2}(t_k\delta_k)^T H(X_k)(t_k\delta_k)$$

$$\frac{d\phi}{dt} = G(X_k)^T\delta_k + t_k\delta_k^T H(X_k)\,\delta_k = 0$$

$$t_k^* = \frac{-G(X_k)^T\delta_k}{\delta_k^T H(X_k)\,\delta_k}$$

$$t_k^* = \frac{-G(X_k)^T\delta_k}{2(\phi' - \phi - G(X_k)^T\delta_k)} = \frac{1}{2\left[1 - \dfrac{(\phi' - \phi)}{G(X_k)^T\delta_k}\right]} \qquad 128$$

dan diambil $q = 1/t^*$. Jika $q > 10$ maka $q = 10$, jika $q < 2$ maka $q = 2$.

Hasil konvergensi diperkirakan bagus jika $R(\delta_k) \to 1$. Jadi perlu perbaikan ε_k berdasarkan $R(\delta_k)$ pada tingkat yang beralasan. Jika $R(\delta_k)$ tertutup ke tunggal (i.e. $R(\delta_k) \geq 3/4$) berarti $\phi(X_k+\delta_k) < \phi(X_k)$ merupakan langkah berhasil maka ε_k dinaikkan dengan mereduksi λ menjadi λ/q. Dan jika $\lambda_{baru} < \dfrac{1}{\|H^{-1}\|_2}$ maka $\lambda = 0$, untuk kekonvergenan kuadratis. Jika $R(\delta_k)$ tidak tertutup ke tunggal (i.e. $R(\delta_k) < 1/4$) maka ε_k harus diturunkan dengan menaikkan λ menjadi $q\lambda$, dimana $2 \leq q \leq 10$. Selanjutnya jika untuk $1/4 \leq R(\delta_k) \leq 3/4$ maka harga λ tidak berubah. Tetapi bila $R(\delta_k) < 0$ maka diambil $R(\delta_k) = 0$ dan membatalkan persamaan (119).

3.2.5 KONVERGENSI METODE MARQUARDT

Ambil titik X_o aproksimasi awal untuk meminimumkan ϕ, menghasilkan barisan dari aproksimasi untuk titik meminimalkannya adalah :

$$X_{k+1} = X_k - t_k[\ J(X_k)^T\ J(X_k) + \lambda_k I\ ]^{-1} J(X_k)^T F(X_k) \qquad 129$$

dimana λ_k adalah barisan dari konstanta real non negatif.

t_k adalah panjang langkah , biasanya diambil $t_k = 1$. Jika $\lambda_k = 0$ dan $J(X_k)$ mempunyai rank penuh dalam persekitaran terbuka dari titik penyelesaian yang dikehendaki, katakan X^* untuk semua k maka menghasilkan metode Gauss Newton dengan iterasi

$$X_{k+1} = X_k - t_k[\ J(X_k)^T\ J(X_k)\ ]^{-1} J(X_k)^T F(X_k) \qquad 130$$

Metode Marquardt menghendaki menentukan nilainya jakobian dari F pada setiap iterasinya. Kita akan lebih nyata mendekati dari aproksimasi $J(X_k)$ dengan korespondensi matriks dari beda hasil bagi. Jadi disatu pihak bisa dengan :

$$J(X) = \partial_i F(X) = \partial\ F(X)/\partial x_i , \qquad 131$$

disisi lain menggunakan :

$$\delta F_i (X,h) = \frac{F(X,h_i U_j) - F(X)}{h_i},\ h_i \neq 0,\ j=1,2,\ldots,N \qquad 132$$

Ambil h_i bilangan real dan ambil $\Delta F(X,h)$ didefinisikan matriks dengan memiliki elemen ke-i dan ke-j.

dimana U_j adalah vektor satuan ke-j. Kita definisikan aproksimasi jakobian dari F:

$$\Delta_i F(X,h) = \begin{cases} \delta_i F(X,h) & , h_i \neq 0 \\ \partial_i F(X) & , h_i = 0 \end{cases} \qquad 133$$

Kita pikirkan memenuhi analogi beda hingga dari algoritma Marquardt:

$$X_{k+1} = X_k - t_k [h_k^{-2} \Delta F(X_k,h_k)^T \Delta F(X_k,h_k) + \lambda_k I\ ]^{-1} h_k^{-1} \Delta F(X_k,h_k)^T F(X_k)$$

$$X_{k+1} = X_k - t_k h_k [\Delta F(X_k,h_k)^T \Delta F(X_k,h_k) + h_k^2 \lambda_k I\ ]^{-1} \Delta F(X_k,h_k)^T F(X_k) \qquad 134$$

Selanjutnya kita ketahui bahwa jika λ_k dan $|\ h_k|$ berorde $O(\|F(X_k)\|_2)$ maka beda hingga Marquardt konvergen kuadratik bilamana $\| F(X_k) \|_2 \xrightarrow[k]{} 0$.

Barisan $\{\lambda_k\}$ nonnegatif dapat dipilih sedemikian sehingga bentuk $\phi(X_k)$ barisan monoton turun. Jika kita buat beberapa asumsi standart, sebagai contoh, bahwa level himpunan dari ϕ terbatas dan kita berhati-hati dalam pemilihan $\{\ \lambda_k\}$ maka suatu subbarisan dari $\{X_k\}$ adalah konvergen menuju titik ekstrim , katakanlah X^* dari ϕ. Jika metode adalah konvergen lokal, maka diberikan syarat subbarisan konvergen cukup diambil tertutup menuju X^*, barisan selanjutnya akan konvergen kepadanya.

Definisi 66

Fungsi F(.) didefinisikan dalam open konveks $D_o \subseteq R^N$ atau $F : D_o \longrightarrow R^N$ dikatakan memenuhi kondisi Lipschitz dengan konstanta $k > 0$ jika

$$\| F(X) - F(Y) \|_2 \leq k \| X - Y \|_2^2, \qquad 135$$

untuk semua $X,Y \in D_o$, dinotasikan $F \in Lip\ (k)$.

Teorema 23

Jika $F : D_o \longrightarrow R^N$ adalah fungsi Lipschitz , maka F adalah kontinu uniform pada D_o.

Bukti :

Jika pertidaksamaan (135) terpenuhi, maka dengan mengambil $\delta(\varepsilon) = \varepsilon/k$ dan $\| X - Y \|_2 < \delta(\varepsilon)$ akan mengakibatkan bahwa $\| F(X) - F(Y) \|_2 < k(\varepsilon/k)$

$$\| F(X) - F(Y) \|_2 < \varepsilon \qquad 136$$

Lemma 3

Ambil D_o himpunan open konveks dan ambil $k \geq 0$. Jika untuk semua $X, Y \in D_o$,

$$\| J(X) - J(Y) \|_2 \leq k \| X - Y \|_2,$$

maka memenuhi pertidaksamaan untuk setiap $X, Y \in D_o$:

i. $\| J(X)^T - J(Y)^T \|_2 \leq k \| X - Y \|_2$,

ii. ada konstanta c dan c' nonnegatif sedemikian sehingga :

$$\| J(X) - J(Y) \|_1 \leq c.k \| X - Y \|_1$$

dan $\| A \|_2 \leq c' \| A \|_1$,

untuk suatu matriks rectangular real A.

iii $\| F(X) - F(Y) - J(Y)(X - Y) \|_1 \leq c.k/2 \| X - Y \|_1^2$,

Bukti :

i. Pandang $\| A \|_2 = \| A^T \|_2$ untuk matriks rectangular A.

sehingga $\| J(X) \|_2 = \| J(X)^T \|_2$, dan

$\| J(Y) \|_2 = \| J(Y)^T \|_2$ dan

maka

$$\|J(X) - J(Y)\|_2 = \| J(X) \|_2 + \|J(Y)\|_2 - 2\| J(X) \|^{1/2} \|J(Y)\|_2^{1/2}$$

$$= \| J(X)^T \|_2 + \|J(Y)^T\|_2 - 2\| J(X)^T \|_2^{1/2} \|J(Y)^T\|_2^{1/2}$$

$$= \|J(X)^T - J(Y)^T\|_2$$
$$\leq k \| X - Y \|_2$$

ii. Pandang vektor jarak l_1 dan l_2 dan c, c' konstanta nonnegatif

$$\| J(X) - J(Y) \|_2 \leq k \| X - Y \|_2,$$
$$\| J(X) - J(Y) \|_2 \leq c \| X - Y \|_2,$$

maka

$$\|J(X) - J(Y)\|_2 . \|J(X) - J(Y)\|_2 \leq c.k \| X - Y \|_2 . \| X - Y \|_2$$
$$\|J(X) - J(Y)\|_1 \leq c.k \| X - Y \|_1$$

selanjutnya

ambil c adalah akar kuadrat positif dari nilai eigen terbesar dari A^TA, sehingga $\| A \|_2 \geq c = \frac{1}{c'} > 0$

karena

$$\| A \|_2 = \| A \|_2$$
$$\| A \|_2 . \| A \|_2 = \| A \|_2 . \| A \|_2$$
$$c \; . \| A \|_2 \leq \| A \|_1$$
$$\frac{1}{c'} . \| A \|_2 \leq \| A \|_1$$
$$\| A \|_2 \leq c' \| A \|_1$$

iii. Pandang

$$F(X) - F(Y) - J(Y)(X - Y) =$$
$$= \int_0^1 \{J [Y + (X-Y)s] + J(Y)\} \, ds \, (X - Y)$$

ambil normanya :

$$\| F(X) - F(Y) - J(Y)(X - Y)\|_2$$
$$= \left\| \int_0^1 \{J [Y + (X-Y)s] + J(Y)\} \, ds \, (X - Y) \right\|_2$$
$$\leq \| X-Y \|_2 \left\| \int_0^1 \{J [Y + (X-Y)s] + J(Y)\} \, ds \right\|_2$$
$$\leq \| X-Y \|_2 \int_0^1 \left\| \{J [Y + (X-Y)s] + J(Y)\} \right\|_2 ds$$

$$\leq \| X-Y \|_2 \int_0^1 k \| X-Y \|_2 \; s \; ds$$

$$\leq \frac{1}{2} \| X-Y \|_2 \; k \; \| X-Y \|_2 \; s \Big|_0^1$$

$$\leq \frac{1}{2} \; k \; \| X-Y \|_2^2$$

sehingga

$$\| F(X) - F(Y) - J(Y)(X - Y)\|_2 \leq \frac{k}{2} \| X-Y \|_2^2$$

maka analog bukti (ii) maka

$$\| F(X) - F(Y) - J(Y)(X - Y)\|_1 \leq \frac{c.k}{2} \| X-Y \|_1^2$$

Proses iterasi suatu saat harus dihentikan setelah dicapai harga yang diinginkan.

Theorema 24

Ambil $A \neq \{\ \}$ himpunan tertutup dalam R^N, ambil $\Omega \subseteq A$ himpunan solusi. Ambil $D : A \longrightarrow A$ pemetaan titik kehimpunan. Diberikan $X_1 \in A$, barisan $\{X_k\}$ dihasilkan secara iterasi yang memenuhi :

Jika $X_k \in \Omega$ maka berhenti;

yang lain, ambil $X_{k+1} \in D(X_k)$,

ganti k dengan k+1, dan diulangi. 137

Misalkan bahwa barisan $X_1, X_2, \ldots, X_k$ dihasilkan olehalgoritma dicapai dalam sup himpunan kompak dari A, dan misalkan terdapat fungsi kontinu ϕ, disebut fungsi menurun, sedemikian sehingga $\phi(Y) < \phi(X)$ jika $X \notin \Omega$ dan $Y \in D(X)$. Jika pemetaan D tertutup atas komplemen dari Ω, maka salah satu algoritma berhenti dalam langkah hingga titik dalam Ω atau menghasilkan barisan infinit $\{X_k\}$ sedemikian sehingga:

i. setiap sub barisan dari $\{X_k\}$ konvergen ke limit dalam Ω, yaitu semua akumulasi titik dari $\{X_k\}$ sepanjang Ω.

ii. $\phi(X_k) \longrightarrow \phi(X)$ untuk semua $X \in \Omega$.

Bukti:

Jika pada suatu iterasi dihasilkan titik $X_k \in \Omega$, maka algoritma berhenti. Sekarang misalkan bahwa dihasilkan barisan $\{X_k\}$ infinit. Kita ambil $\{X_k\}_K$ sub himpunan konvergen dengan limit $X \in A$. selama ϕ kontinu, maka untuk $k \in K$, $\phi(X_k) \longrightarrow \phi(X)$. Jadi berikan $\varepsilon > 0$, terdapat $k \in K$ sedemikian sehingga

$$\phi(X_k) - \phi(X) < \varepsilon, \text{ untuk } k \geq \kappa \text{ dengan } k \in K.$$

Terutama untuk $k = \kappa$, kita dapatkan

$$\phi(X_k) - \phi(X) < \varepsilon \qquad 139$$

Sekarang ambil $k > \kappa$. Selama ϕ adalah fungsi descent, $\phi(X_k) < \phi(X_\kappa)$ dan dari persamaan (139) kita berikan

$$\phi(X_k) - \phi(X) = \phi(X_k) - \phi(X_\kappa) + \phi(X_\kappa) - \phi(X) < 0 + \varepsilon = \varepsilon$$

$$\phi(X_k) - \phi(X) < \varepsilon.$$

Selama ini benar untuk $\forall\ k > \kappa$, dan ambil sebarang $\varepsilon > 0$, maka

$$\lim_{k \longrightarrow 0} \phi(X_k) = \phi(X) \qquad 140$$

Sekarang diperlihatkan bahwa $X \in \Omega$. Dengan kontradiksi, misalkan bahwa $X \notin \Omega$, dan pikirkan barisan $\{X_{k+1}\}_K$. Barisan ini dicapai dalam subbarisan kompak A dan sedemikian sehingga sub barisan $\{X_{k+1}\}_{\bar{K}}$ konvergen dengan limit X^* dalam A. Catatan persamaan (140), ternyata bahwa

$$X_{k+1} \longrightarrow X^*, \text{ maka } X^* \in D(X).$$

Oleh karena itu $\phi(X^*) < \phi(X)$, kenyataan ini kontradiksi bahwa $\phi(X^*) = \phi(X)$. Jadi $X \in \Omega$.

Selanjutnya terlihat bahwa dari persamaan (140);

$$\lim_{k \longrightarrow 0} \phi(X_k) = \phi(X)$$

maka $\phi(X_k) \longrightarrow \phi(X)$, untuk suatu $X \in \Omega$. ∎

Akibat 2

Dibawah asumsi teorema diatas, jika Ω adalah singleton $\{X^*\}$, maka seluruh barisan $\{X_k\}$ konvergen menuju X^*.

Bukti :

Dengan kontradiksi, misalkan terdapat $\varepsilon > 0$, dan barisan $\{X_k\}_K$ sedemikian sehingga

$$\| X_k - X^* \|_2 > \varepsilon \text{ , untuk } k \in K. \qquad 141$$

Catatan bahwa terdapat $K' \subset K$ sedemikian sehingga $\{X_k\}_{K'}$ mempunyai limit X. Dengan theorema (94) bagian satu, $X' \in \Omega$. Tetapi $\Omega = \{X^*\}$, dan jadi $X' = X^*$. Oleh karena itu $X_k \longrightarrow X^*$ untuk $k \in K'$, ini kontradiksi dengan persamaan (141). ∎

Algoritma kita hentikan jika kita telah mencapai titik dalam himpunan solusi Ω. Dalam banyak hal, kekonvergenan ke titik dalam Ω hanya dalam keadaan limit, dan kita harus berhenti dengan beberapa aturan praktis untuk terminasi prosedur iterasi. Pertama kali memberikan batas kesalahan yang dikehendaki ($\varepsilon > 0$) dan maksimum iterasi yang dikehendaki (MAXIT) bilangan integer positif, selanjutnya definisikan :

1. $\| X_{k+MAXIT} - X_k \|_2 < \varepsilon$ 142

   Algoritma berhenti jika jarak variabel dalam pemetaan D telah bergerak sampai MAXIT kali adalah lebih kecil dari pada ε

2. $$\frac{\| X_{k+1} - X_k \|_2}{1 + \| X_k \|_2} < \varepsilon \qquad 143$$

   Algoritma dibatasi jika jarak relatif variabel bergerak selama iterasi memberikan harga lebih kecil dari pada ε

3. $$\| \phi(X_k) - \phi(X_{k+MAXIT}) \|_2 < \varepsilon \qquad 144$$

   Algoritma berhenti jika keseluruhan perbaikan dalam harga fungsi menurun setelah MAXIT kali dari pemetaan D adalah lebih kecil dari pada ε

4. $$\frac{\| \phi(X_k) - \phi(X_{k+1}) \|_2}{1 + \| \phi(X_{k+1}) \|_2} < \varepsilon \qquad 145$$

   Algoritma dibatasi jika perbaikan relatif dalam harga fungsi descent selama iterasi memberikan harga lebih kecil dari pada ε

Berikut ini, kita berikan strategi untuk pembaharuan λ.

Ambil $2 \le q \le 10$

Ambil harga awal $\lambda = 10^{-4}$ (bilangan positif kecil)

Definisikan λ_{k-1} harga λ dari iterasi awal.

Hitung $\phi(X_{k-1})$ dan $\phi(X_{k-1}/q)$

i. jika $\phi(\lambda_{k-1}/q) \le \phi_k$, ambil $\lambda_k = \lambda_k/q$

ii. jika $\phi(\lambda_{k-1}/q) > \phi_k$ dan $\phi(\lambda_{k-1}) \le \phi_k$, ambil $\lambda_k = \lambda_{k-1}$

iii. jika $\phi(\lambda_{k-1}/q) > \phi_k$ dan $\phi(\lambda_{k-1}) > \phi_k$,

perbesar λ dengan kalikan q hingga untuk suatu nilai w terkecil dipenuhi

$\phi(\lambda_{k-1}q^{w}) \le \phi_k$, ambil $\lambda_k = \lambda_{k-1}\, q^{w}$.

Dalam hal ini kebijaksanaan yang diambil untuk menurunkan/menaikkan λ parameter Marquardt adalah sebagai berikut :

i. jika langkah berhasil ($\phi(X_k+\delta_k) < \phi(X_k)$ maka turunkan λ_k dengan mengambil $\lambda_{k+1} = \lambda_k/10$

ii. jika $\lambda_k < 10^{-20}$ maka diambil $\lambda_{k+1} = 10^{-20}$, dimana $\lambda_k = 0$ tidak diijinkan.

iii. jika langkah tak berhasil ($\phi(X_k+\delta_k) \ge \phi(X_k)$ maka menaikkan λ_k dengan mengambil $\lambda_{k+1} = 10\lambda_k$.

iv. Suatu saat aproksimasi Hessian tidak definit positif atau det(H) < 10^{-6}, maka $\lambda_{k+1} = 100\lambda_k$, untuk menetapkan kedefinit - positifan dengan faktorisasi.

Selanjutnya untuk memperjelas implementasi metode Marquardt akan dipresentasikan dalam algoritma dan flow chart.

3.3 ALGORITMA METODE MARQUARDT DAN IMPLEMENTASINYA

Algoritma Marquardt menghendaki mencari $\lambda \geq 0$ sedemikian sehingga $H_k + \lambda I$ adalah definit positif dan menyelesaikan :

$$(H_k + \lambda I)\,\delta = -G_k,$$

untuk menghitung δ_k. Skalar λ dalam model digunakan sebagai parameter kontrol dalam iterasi yang harus dihitung sedemikian sehingga $\| \delta_k \|_2 = \varepsilon_k$ dicapai, sehingga apabila menaikkan λ menyebabkan $\| \delta \|_2$ menurun.

Adapun algoritma secara keseluruhan adalah sebagai berikut:

1. Memasukkan jumlah variabel dan harga variabel
2. Memasukkan kontrol parameter
3. Menghitung residual $F(X_k)$
4. Menghitung $\phi(X_k)$
5. Menghitung jakobian $J(X_k)$
6. menghitung Matriks normal $H(X_k) = J(X_k)^T J(X_k)$
7. Normalisasi faktor skala menjadi $D(X_k)$

$$D(X_k) = \frac{H(X_k)}{\| H(X_k) \|_2}$$

8. Menghitung gradien $G(X_k) = \nabla \phi(X_k) = J(X_k)^T F(X_k)$ dan panjangnya $G1 = \| G(X_k) \|_2$
9. Menghitung dan membandingkan gradien dari jakobian terhadap aproksimasi beda hasil bagi yaitu :

$$G \text{ dengan } \frac{\phi(X_k+h) - \phi(X_k)}{h}$$

10. Menghitung iterasi awal $k = 0$ dan memasukkan titik

untuk iterasi baru, lalu menaikkan iterasi k =k+1, dan menyimpan nilai ϕ yang terakhir.

11. Kebijaksanaan untuk menaikkan / menurunkan parameter λ pada tiap awal iterasi :
    i. jika langkah berhasil ($\phi(X_k+\delta_k) < \phi(X_k)$ maka turunkan λ_k dengan mengambil $\lambda_{k+1} = \lambda_k/10$
    ii. jika $\lambda_k<10^{-20}$ maka diambil $\lambda_{k+1}= 10^{-20}$, dimana $\lambda_k=0$ tidak diijinkan.
    iii. Jika langkah tak berhasil ($\phi(X_k+\delta_k) \geq \phi(X_k)$ maka menaikkan λ dengan mengambil $\lambda_{k+1}= 10\lambda_k$.
    iv. Suatu saat aproksimasi Hessian tidak definit positif atau det(H) < 10^{-6}, maka $\lambda_{k+1}= 100\lambda_k$, untuk menetapkan kedefinit-positifan dengan faktorisasi.
12. Menghitung jakobian, gradien dan panjangnya, dan disimpan pada matriks normal dalam $H(X_k)$.
13. Mencetak $\phi(X_k)$,X_k, $G(X_k)$
14. Penambahan parameter Marquardt λ pada matriks Normal $H(X_k)$, dimana $H = H + \lambda D$
15. Faktorisasi Choleski LDL^T untuk matriks Hessian H, yaitu :

$$J^TJ + \lambda D = LDL^T$$

16. Mengetest kondisi faktorisasi LDL^T jika tidak berhasil, maka menaikkan parameter Marquardt dengan $\lambda_{k+1}= 100\lambda_k$, lalu menghitung dan menyimpan matriks normal dalam H, meloncat ke langkah (14) untuk revisi dan refaktor H,jika berhasil maka

kontinu

17. Mengganti sisi ruas kanan dengan : $-G(X_k)$
18. Menghitung langkah δ_k dalam E_k,

    dimana $\delta_k = -(H + \lambda D^2)^{-1}G$
19. Menghitung langkah δ_k dalam derajad gradien

$$\text{Cos } \gamma = - \frac{\delta_k^T G}{\| \delta_k \|_2 \| G \|_2}$$

    dan mencetak langkah dalam derajad gradien.
20. Mengambil langkah baru : $X_{k+1} = X_k + \delta_k$
21. Menghitung residual $F(X_{k+1})$
22. Menghitung $\phi(X_{k+1}) = \frac{1}{2} \| F(X_{k+1}) \|^2$
23. Mengetest kondisi bahwa $\phi(X_{k+1}) < \phi(X_k)$

    jika ya maka melampat kelangkah (26)

    jika tidak maka kontinu
24. Menghitung $X_{k+1} = X_k - \delta_k$

    lalu diambil $\delta_k = \delta_k/4$, $m_3 = m_3 + 1$.
25. Mengetest kondisi apakah $M_3 < 11$

    jika tidak maka mencetak "ukuran langkah terlalu kecil"

    untuk berhenti, melompat ke langkah (32).
26. Mengetest iterasi < maksimum iterasi yang dikendaki

    jika ya maka melompat ke langkah (27).

    jika tidak maka cetak " stop pada pemberian limit",

    lalu melompat ke langkah (31).
27. Mengetest kekonvergenan dari ϕ

$$\text{jika } \frac{\|\phi(X_k) - \phi(X_{k+1})\|_2}{1 + \| \phi(X_{k+1}) \|_2} > \varepsilon \text{ maka ke langkah (10)}$$

biasanya $\varepsilon = 10^{-4}$

jika tidak maka kontinu

28. Test kekonvergenan setiap X_k

$$\text{jika } \frac{\| X_{k+1} - X_k \|_2}{1 + \| X_k \|_2} > \varepsilon \quad \text{maka ke langkah (10)}$$

jika tidak maka kontinu

29. k = k + 1; Mencetak "konvergensi dan solusinya adalah"
30. Menghitung Jakobian
31. Menghitung gradien
32. Mecetakkan ϕ, X, G pada titik pemberhentian.
33. Mencetak residual
34. mencetak total evaluasi fungsi.
35. Stop.

Adapun subroutine utama dalam program optimasi kuadrat terkecil dengan metode Marquardt adalah sebagai berikut :

No.	NAMA	BARIS
1.	Masukkan jumlah variabel dan harga variabel	1200-1270
2.	Masukkan/revisi kontrol parameter..........	2280-1360
3.	Algoritma optimasi utama....................	1400-2490
4.	Tampilkan fungsi, gradien dan variabel......	2500-2580
5.	Hitung dan simpan matriks normal dalam H()..	2590-2730
6.	Hitung gradien dan panjangnya...............	2740-2870
7.	Penambahan parameter Marquardt untuk matriks normal....................................	2860-2970
8.	Tampilkan data sampel dari baris 400-600....	2980-3080

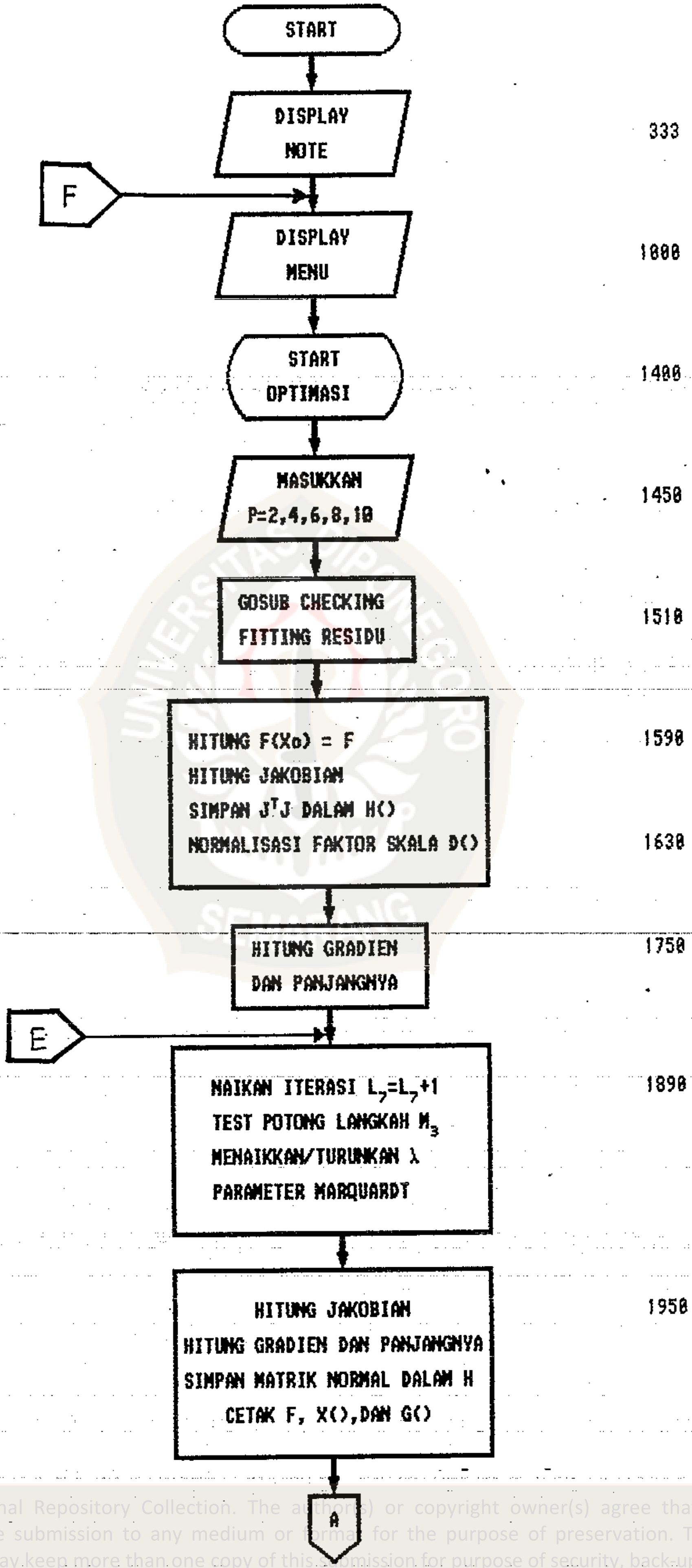
9. Faktorisasi LDL^T dari Hessian dalam pembaharuan solusi **X**.............................3090-3360

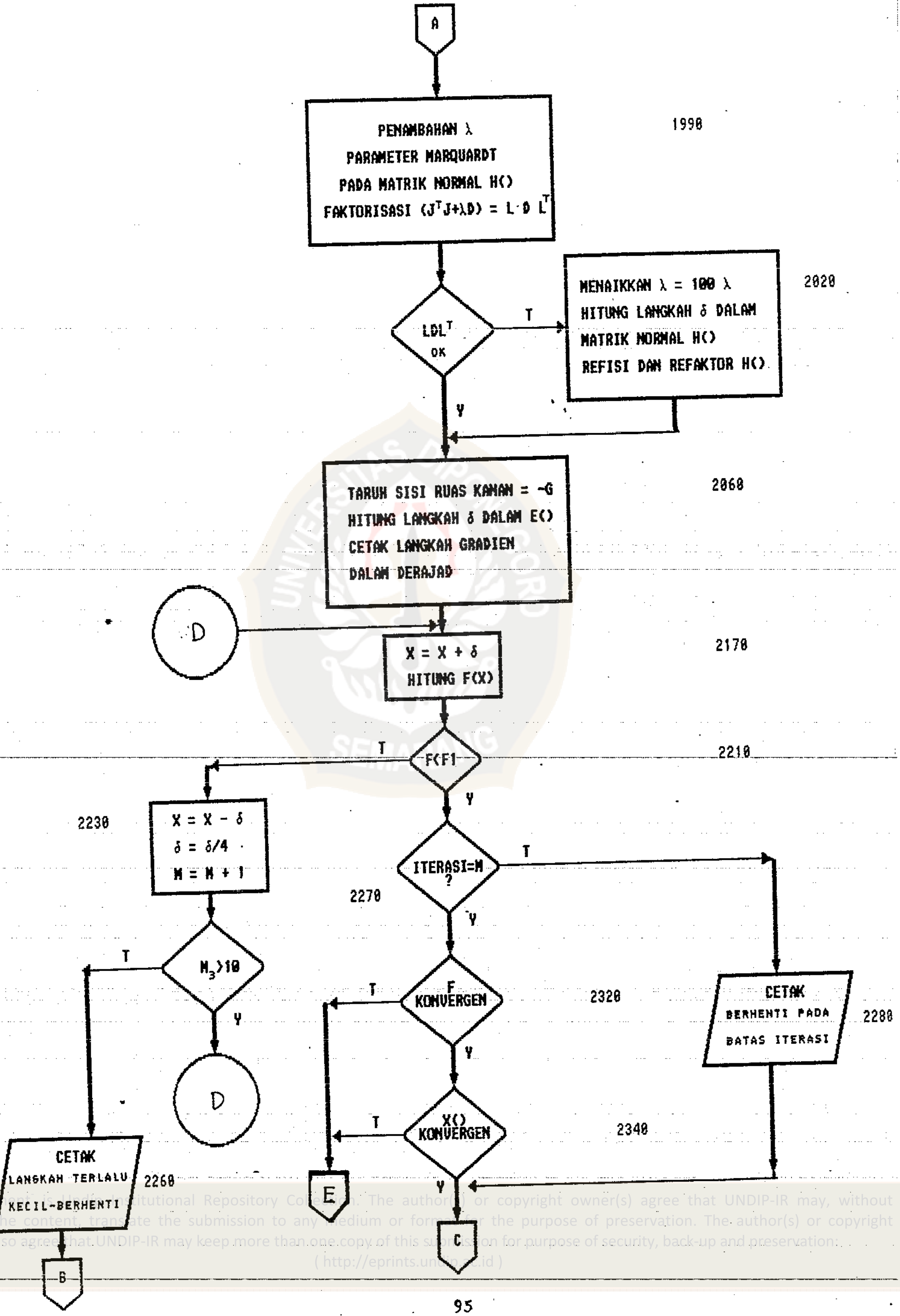
10. Penyelesaian untuk langkah δ dalam $(J^TJ + \lambda D^2)\ \delta = -\ G$..........................3370-3600

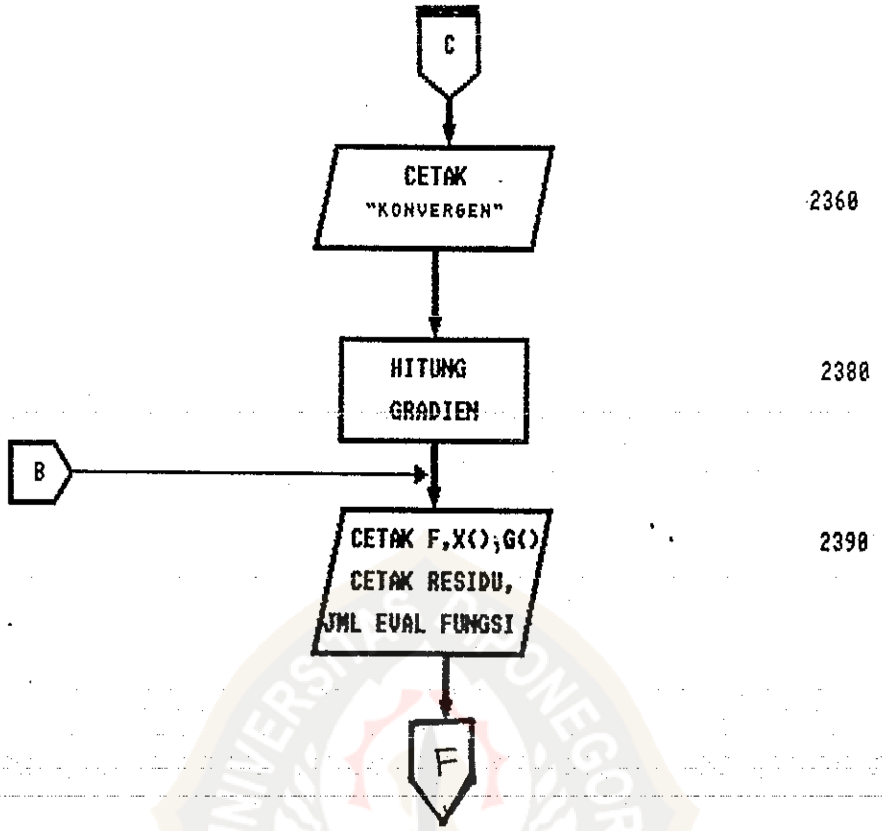
11. Residual f untuk ϕ dan G 5000-6999

12. Matriks Jakobian $J = [\ \nabla_j f_k\ ]$..............7000-8999

Selanjutnya disajikan folw Chart untuk gambaran secara lengkap tentang langkah-langkah dalam optimasi dengan metode Marquardt, sehingga memudahkan untuk membuat programnya.


START
DISPLAY NOTE
333
F
DISPLAY MENU
1000
START OPTIMASI
1400
MASUKKAN P=2,4,6,8,10
1450
GOSUB CHECKING FITTING RESIDU
1510
HITUNG F(Xo) = F
HITUNG JAKOBIAN
SIMPAN J^TJ DALAM H()
NORMALISASI FAKTOR SKALA D()
1590
1630
HITUNG GRADIEN DAN PANJANGNYA
1750
E
NAIKAN ITERASI $L_7=L_7+1$
TEST POTONG LANGKAH M_3
MENAIKKAN/TURUNKAN λ
PARAMETER MARQUARDT
1890
HITUNG JAKOBIAN
HITUNG GRADIEN DAN PANJANGNYA
SIMPAN MATRIK NORMAL DALAM H
CETAK F, X(),DAN G()
1950
A


A
PENAMBAHAN λ
PARAMETER MARQUARDT
PADA MATRIK NORMAL H()
FAKTORISASI $(J^TJ+\lambda D) = L \cdot D\ L^T$
1990
LDL^T OK
T
MENAIKKAN λ = 100 λ
HITUNG LANGKAH δ DALAM
MATRIK NORMAL H()
REFISI DAN REFAKTOR H()
2020
Y
TARUH SISI RUAS KANAN = -G
HITUNG LANGKAH δ DALAM E()
CETAK LANGKAH GRADIEN
DALAM DERAJAD
2060
D
X = X + δ
HITUNG F(X)
2170
F<F1
T
2210
Y
2230
X = X - δ
δ = δ/4
M = M + 1
ITERASI=M ?
T
2270
Y
M_3>10
T
Y
D
F KONVERGEN
T
Y
2320
CETAK
BERHENTI PADA
BATAS ITERASI
2280
X() KONVERGEN
T
2340
Y
CETAK
LANGKAH TERLALU
KECIL-BERHENTI
2260
E
C
B


C
CETAK
"KONVERGEN"
2360
HITUNG
GRADIEN
2380
B
CETAK F,X();G()
CETAK RESIDU,
JML EVAL FUNGSI
2390
F

```
10 REM ***************************************************************
11 REM *    IMPLEMENTASI METODE MARQUARDT DALAM OPTIMASI NONLINEAR    *
12 REM *                                                              *
13 REM *               PROGRAMMER   : PRIYO SIDIK SASONGKO            *
14 REM *               NIM          : J 101 90 0374                   *
15 REM *               JURUSAN      : MATEMATIKA / MIPA -UNDIP        *
16 REM *                                                              *
17 REM ***************************************************************
110 REM
119 REM **************************************************************
120 REM *                  VARIABEL YANG DIGUNAKAN                    *
121 REM *                                                             *
130 REM * A( , ) : Matrik Jakobian                                    *
140 REM * D()    : Vektor untuk matrik penskalaan diagonal Marquardt  *
150 REM * D1     : Determinan dari Faktorisasi LDLT                   *
160 REM * E()    : Vektor langkah penyelidikan                        *
170 REM * F      : Setengah jumlahan residual kuadrat                 *
180 REM * F1     : Penyimpanan harga F untuk perbandingan menurun     *
190 REM * FNACX  : Invers fungsi Cosinus                              *
200 REM * G()    : Gradien dari F                                     *
210 REM * G1     : Panjang dari gradien                               *
220 REM * H()    : Vektor simpanan dari aproksimasi matrik Hessian    *
221 REM *          berdimensi N(N+1)/2.                               *
230 REM * K2     : Hitung jumlah evaluasi  F                          *
240 REM * K7     : Eksponen pangkat  P untuk residual                 *
250 REM * M      : Counter iterasi                                    *
260 REM * M3     : Jumlah data sampel yang dibaca dari statemen data  *
270 REM * N      : Jumlah variabel, yang diijinkan N <= 20            *
280 REM * R()    : Residual , berdimensi N                            *
290 REM * S( , ) : S(i,1) adalah sampel independen                    *
291 REM *          S(i,2) adalah sampel dependen                      *
300 REM * V      : Parameter Marquardt                                *
310 REM * X()    : Vektor variabel, berdimensi N                      *
311 REM *                                                             *
312 REM **************************************************************
320 DEFDBL A-H , O-Z : REM CATATAN BAHWA P ADALAH KETEPATAN TUNGGAL
330 DEFINT I-N
331 OPTION BASE 1
332 CLS : KEY OFF
333 PRINT "OPTIMASI KUADRAT TERKECIL"
334 PRINT "NOTES :"
335 PRINT "1. APLIKASI PADA CONTOH SOAL"
336 PRINT "2. IF BERHENTI KEMBALI KE - 999
337 PRINT "3. PENGGUNA HARUS KE SUBROUTINE 5000 UNTUK HITUNG RESIDUAL"
338 PRINT "      SUBROUTINE 7000 UNTUK JACOBIAN MATRIK DAN SAMPEL DATA"
339 PRINT "      PADA BARIS 400-600 JIKA DITANYAKAN"
340 PRINT
345 S6$ = "######.#########"
350 DEF FNACX(X) =1.570796 - ATN(X/SQR(1-X*X))
360 M1 = 50 : E1 = .0001 : I7 = 1 : L1 = 0 : M3 =0 : V = .001
365 M7 = 0
390 DIM X(20), G(20),H(210),E(20),A(40,20),S(40,2),D(20),R(40)
410 READ N$ : PRINT  "BEKERJA DENGAN HIMPUNAN DATA :" ;N$ :PRINT
420 READ M7
430 FOR K = 1 TO M7 : READ S(K,1) : NEXT K
440 FOR K = 1 TO M7 : READ S(K,2) : NEXT K
450 GOTO 1160 : REM - MENU PILIHAN
```

```
480 DATA "DUMMY" : REM NAMA HIMPUNAN DATA
485 DATA 5
490 DATA 1,2,3,4,5  : REM HARGA DATA INDEPENDEN
500 DATA 1.1,2.2,3.3,4.4,5.5 : REM HARGA DATA DEPENDEN
510 REM MEMASUKKAN KEMBALI JIKA ANGKA PERINTAH SALAH DAN CONTINU
999 CLS : K2=0 : REM FUNGSI AWAL YANG DI EVALUASI
1000 PRINT "********************** MENU PILIHAN ************************"
1005 PRINT "*                                                         *"
1010 PRINT "*                  1. MASUKAN VARIABEL AWAL               *"
1020 PRINT "*                  2. REVISI CONTROL PARAMETER            *"
1030 PRINT "*                  3. START OPTIMASI                      *"
1040 PRINT "*                  4. KELUAR                              *"
1050 PRINT "*                  5. SPARE                               *"
1060 PRINT "*                  6. DISPLAY PASANGAN DATA               *"
1064 PRINT "*                                                         *"
1065 PRINT "***********************************************************"
1070 PRINT
1080 PRINT "INPUT  BILANGAN PERINTAH :"; :INPUT S$
1090 K = LEN(S$) : IF K = 0 THEN GOTO 999
1100 K = ASC(S$)
1110 IF K<48 OR K>57 THEN GOTO 999
1120 K = VAL(S$)
1130 IF K=0 THEN K=15
1140 IF K>20 THEN  GOTO  999
1150 ON K GOSUB 1220,1290,1420,1390,999,3000
1160 PRINT "TEKAN SEBARANG TOMBOL UNTUK KONTINU - READY :";
1170 INPUT S4$
1180 IF S4$<>"" THEN BEEP
1190 GOTO 999
1200 REM ***********************************************************
1210 REM *                    MASUKAN VARIABEL                      *
1211 REM ***********************************************************
1220 PRINT "JUMLAH VARIABEL :" ; : INPUT N
1230 PRINT "MASUKKAN VARIABEL AWAL X(I) :"
1240 FOR I = 1 TO N
1250 PRINT "    X(";I;") = "; :INPUT X(I)
1255 XA(I)=X(I)
1260 NEXT I
1270 RETURN
1280 REM ***********************************************************
1290 REM *               REVISI KONTROL PARAMETER                   *
1291 REM ***********************************************************
1300 PRINT "MAKSIMUM ITERASI ( DEFAULT = 50) :" ;: INPUT S4$
1310 M1=50 : IF S4$<>"" THEN  M1=VAL(S4$)
1320 PRINT "KRITERIA STOP ( DEFAULT = 0.0001) :" ;: INPUT S4$
1330 E1=.0001 : IF S4$<>"" THEN E1= VAL(S4$)
1340 PRINT "PRINT SETIAP ITERASI (DEFAULT=1)" ; :INPUT S4$
1350 I7=1 : IF S4$<>"" THEN I7=VAL(S4$)
1360 RETURN
1370 REM ***********************************************************
1380 REM *                      STOP PROGRAM                        *
1381 REM ***********************************************************
1390 KEY ON :PRINT " AKHIR PROSES " : END
1400 REM ***********************************************************
1410 REM *               ALGORITMA PROGRAM UTAMA                    *
1415 REM ***********************************************************
1420 IF N>0 THEN GOTO 1450
```

```
1430 PRINT "-N JUMLAH VARIABEL TIDAK DIKEHENDAKI : GUNAKAN PERINTAH #1-"
1440 RETURN
1450 PRINT "EKSPONEN P(2,4,6,8 OR 10 ) =" ;:INPUT K7
1460 FOR I = 1 TO M : REM - A(I,J)= 0 MATRIK JACOBIAN
1470 FOR J = 1 TO N
1480 A(I,J) = 0
1490 NEXT J
1500 NEXT I
1505 REM ****************************************************************
1506 REM *              GOSUB 5000 : HITUNG RESIDU PERTAMA                *
1508 REM ****************************************************************
1510 GOSUB 5000
1511 PRINT "PENGGUNAAN SEKUMPULAN BILANGAN DARI SAMPEL M = ";M;
1520 PRINT "DALAM SUBROUTINE 5000"
1530 PRINT "    ADALAH KONSISTEN  DENGAN MASALAH INI (y/n) ";
1540 INPUT S4$ : IF S4$ ="N" THEN RETURN
1544 REM ****************************************************************
1545 REM *                F DIEVALUASI PERHITUNGAN MENAIK                 *
1546 REM ****************************************************************
1550 K2 = K2 + 1
1560 IF L1=0 THEN  GOTO 1580 : REM - SUBROUTINE RESIDU TIDAK GAGAL
1570 PRINT "SUBROUTINE 5000 TAK BERLAKU UNTUK MENGHITUNG " : RETURN
1574 REM ****************************************************************
1575 REM *            KALKULASI JUMLAHAN RESIDUAL KE-P PERTAMA            *
1576 REM ****************************************************************
1580 F = 0
1590 FOR K = 1 TO M : F = F +R(K)^K7 : NEXT K : F = F/K7
1591 REM ****************************************************************
1592 REM *    GOSUB 7000 : HITUNG JAKOBIAN PERTAMA                        *
1593 REM *    GOSUB 2600 : HITUNG/SIMPAN MATRIK NORMAL DALAM H()          *
1594 REM ****************************************************************
1600 GOSUB 7000
1610 GOSUB 2600
1611 REM ****************************************************************
1620 REM *          AMBIL NORMALISASI FAKTOR SKALA MENJADI D()            *
1621 REM ****************************************************************
1630 L=0 : D2 = 0
1640 FOR J = 1 TO N
1650 FOR I = 1 TO N
1660 IF I<J THEN GOTO 1690
1670 L = L+1
1680 IF I = J THEN D(J)=H(L)
1690 NEXT I
1700 IF D(J)<= 0 THEN D(J) = 1
1710 D2 = D2 + D(J)*D(J)
1720 NEXT J
1730 D2 = SQR(D2)
1731 REM *  NORMALISASI *
1740 FOR J = 1 TO N : D(J)=D(J)/D2 : NEXT J
1741 REM ****************************************************************
1742 REM *     GOSUB 2750 : KALKULASI GRADIEN G() DAN PANJANG G1          *
1743 REM ****************************************************************
1750 GOSUB 2750
1751 REM ****************************************************************
1760 REM *              PERBANDINGAN GRADIEN UNTUK BEDA HINGGA            *
1761 REM ****************************************************************
1770 PRINT " GRADIEN      VIA SUB 7000      VIA PERBEDAAN  "
```

```
1780 FOR J1 = 1 TO N
1790 D5 = .0001 * ABS(X(J1)) : IF D5< .000001 THEN D5=.000001
1800 X(J1) = X(J1)+D5 :GOSUB 5000 : REM - PERTUBED RESIDU
1810 F2 = 0 : FOR I = 1 TO M : F2 = F2+ R(I)^K7 : NEXT I : F2 =F2/K7
1820 PRINT USING "        ######.#########";G(J1),(F2-F)/D5
1830 X(J1) = X(J1)-D5
1840 NEXT J1
1850 PRINT " TEKAN ENTER UNTUK KONTINU --READY " ; :INPUT S4$
1855 REM *************************************************************
1856 REM *      - MENGHITUNG ITERASI AWAL                              *
1857 REM *      - KEMBALI MEMASUKKAN TITIK UNTUK ITERASI BARU          *
1858 REM *      - HITUNG ITERASI MENAIK DAN SIMPAN NILAI F TERAKHIR    *
1859 REM *************************************************************
1860 L7 = 0
1890 L7 = L7 + 1 : F1= F
1895 IF (L7=1) THEN FA=F
1900 IF M3 = 0 THEN V = V/10 : REM LANGKAH TERAKHIR BAIK,V DIREDUKSI
1910 IF V<1D-20  THEN V=1D-20
1920 IF M3<>0 THEN  V = 10*V
1930 M3 = 0
1940 IF L7=1 THEN GOTO 1980
1950 GOSUB 7000
1960 GOSUB 2750
1970 GOSUB 2600
1980 IF ((L7-1) MOD I7) =0 THEN GOSUB 2500
1990 GOSUB 2870
2000 GOSUB 3110
2010 IF N5=0 THEN  GOTO  2060
2020 V = 100*V
2030 GOSUB 2600
2040 GOTO 1990
2045 REM *************************************************************
2046 REM *                GANTI RUAS KANAN DENGAN : - G()              *
2050 REM *************************************************************
2060 FOR I = 1 TO N : E(I)=-G(I) : NEXT I
2070 GOSUB 3390
2071 REM *************************************************************
2080 REM *             KALKULASI LANGKAH KE TINGKAT GRADIEN            *
2081 REM *************************************************************
2090 C2 = 0 : C3 = 0
2100 FOR I = 1 TO N : C2 = C2+G(I)*E(I) : C3=C3+E(I)*E(I) : NEXT I
2110 P1 = -C2/G1/SQR(C3) : IF P1<1 THEN GOTO 2130
2120 P1= 0 : GOTO  2140
2130 P1 = 57.29578 *FNACX(P1)
2140 PRINT "          LANGKAH KE DERAJAD GRADIEN           =";
2150 PRINT USING " ##.#";P1
2151 REM *************************************************************
2160 REM *            AMBIL LANGKAH DENGAN KENAIKAN DALAM E()          *
2161 REM *            GOSUB 5000 : KALKULASI RESIDUAL                  *
2162 REM *************************************************************
2170 FOR I = 1 TO N : X(I) =X(I)+E(I) : NEXT I
2180 GOSUB 5000
2190 K2= K2 +1 : F= 0
2200 FOR K =  1 TO M : F=F+R(K)^K7 : NEXT K : F = F/K7
2210 IF F< F1 THEN  GOTO  2270
```

```
2211 REM ***************************************************************
2220 REM *                 AMBIL KEMBALI UNTUK MENGAKHIRI                *
2221 REM ***************************************************************
2230 FOR I = 1 TO N : X(I) = X(I)-E(I) : E(I) = E(I)/4 : NEXT I
2240 PRINT " ***** POTONG KEMBALI UKURAN LANGKAH DENGAN FAKTOR 4 ***** "
2250 M3 = M3 + 1  : IF M3<11 THEN  GOTO  2170
2260 PRINT " UKURAN LANGKAH TERLALU KECIL - TERMINASI " : GOTO 2390
2270 IF L7<M1 THEN  GOTO  2320 : REM -TIDAK PADA ITERASI MAKSIMUM
2280 PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
2290 PRINT "STOP PADA PEMBERIAN LIMIT DARI ";M1;"ITERASI HASIL ADALAH :"
2300 L7 = L7 + 1 : GOTO 2370
2301 REM ***************************************************************
2310 REM *        TEST KONVERGEN DARI KEDUANYA F DAN SETIAP X(I)       *
2311 REM ***************************************************************
2320 IF ABS(F1-F)/(1+ABS(F1))>E1 THEN  GOTO  1890
2330 FOR  I = 1 TO N
2340 IF ABS(E(I))/(1+ABS(X(I)))>E1 THEN GOTO 1890
2350 NEXT I
2360 L7 =L7 +1 : PRINT " KONVERGEN  : SOLUSI ADALAH :"
2361 REM ***************************************************************
2362 REM *               GOSUB 7000  : AMBIL JAKOBIAN                  *
2363 REM *               GOSUB 2750  : AMBIL GRADIEN                   *
2364 REM *               CETAK F,X, DAN G PADA TITIK PEMBERHENTIAN     *
2365 REM ***************************************************************
2370 GOSUB 7000
2380 GOSUB 2750
2390 GOSUB 2500
2400 PRINT "TEKAN ENTER UNTUK KONTINU " ; : INPUT S4$
2405 REM ***************************************************************
2406 REM *                    CETAK RESIDUAL DAN ANALISA               *
2407 REM ***************************************************************
2410 K=0 : PRINT "RESIDU ADALAH "
2420 FOR I = 1 TO M : PRINT I,R(I)
2430 IF I<21 OR K=1 THEN GOTO 2455
2440 PRINT "TEKAN SEBARANG KUNCI UNTUK KONTINU " ; :INPUT S4$
2450 K = 1
2455 NEXT I
2459 A$ = "    ####.#####^^^^     #####.#####^^^^     ##.##^^^^ "
2460 P$ = "##  ####.######^^^^  #####.######^^^^  #.######^^^^  #.######^^^^"
2461 PRINT
2462 PRINT "                     ANALISIS  MODEL NONLINEAR"
2463 PRINT "=================================================================="
2464 PRINT " NO.      Yi              Yi           R(i)        SIMPANGAN "
2465 PRINT "      OBSERVASI       CALCULASI       RESIDU       BAKU NORMAL"
2466 PRINT "=================================================================="
2467 FOR I = 1 TO M
2468 PRINT USING P$;I;S(I,2);S(I,2)-R(I);R(I);R(I)/SQR(F)
2469 NEXT I
2470 PRINT "=================================================================="
2471 PRINT  "TOTAL BILANGAN UNTUK EVALUASI FUNGSI =";K2
2472 PRINT "HARGA MINIMAL DARI φ(X)              =";F
2473 IF(M=N) THEN P =0 ELSE P=N
2474 PRINT "VARIANSI                             =";F/(M-P)
2475 PRINT "STANDART DEVIASI                     =";SQR(F/(M-P))
2480 PRINT "EXPONEN P = ";K7
2481 PRINT
2482 PRINT "                     ANALISIS HARGA VARIABEL  "
```

```
2483 PRINT "=============================================================="
2484 PRINT " VARIABEL          AWAL          AKHIR       KEPERCAYAAN %      "
2485 PRINT "=============================================================="
2486 FOR I = 1 TO N
2487 PRINT " X(";I;")";
2488 PRINT USING A$;XA(I);X(I);(X(I)-XA(I))*100/XA(I)
2489 NEXT I
2490 PRINT " Q( X )";
2491 PRINT USING A$;FA;F;(F-FA)*100/FA
2492 PRINT "=============================================================="
2495 RETURN
2500 REM *************************************************************
2510 REM *            CETAK FUNGSI , VARIABEL  DAN GRADIEN            *
2511 REM *************************************************************
2520 PRINT " PADA START DARI ITERASI ";L7 ;:PRINT ";";
2530 P1 = F : PRINT " HARGA FUNGSI =";P1
2540 PRINT " I              X(I)        G(I)   "
2550 FOR I = 1 TO N
2560 PRINT I; : PRINT USING S6$;X(I),G(I)
2570 NEXT I
2580 RETURN
2590 REM *************************************************************
2600 REM *          CALCULASI/ SIMPAN MATRIK NORMAL DALAM H()         *
2605 REM *************************************************************
2610 FOR I = 1 TO N*(N+1)/2 : H(I)=0 : NEXT I
2620 FOR K = 1 TO M
2630 L=0
2640 FOR J = 1 TO N
2650 FOR I = 1 TO N
2660 IF I<J THEN GOTO 2690
2670 L = L+1
2680 H(L)=H(L) + A(K,I)*A(K,J)*R(K)^(K7-2)
2690 NEXT I
2700 NEXT J
2710 NEXT K
2720 FOR L = 1 TO N*(N+1)/2 : H(L)=  (K7-1)*H(L) : NEXT L
2730 RETURN
2740 REM *************************************************************
2750 REM *              KALKULASI GRADIEN DAN PANJANGNYA              *
2751 REM *************************************************************
2760 G1=0
2770 FOR I = 1 TO N
2780 G(I)=0
2790 FOR K = 1 TO M
2800 G(I) = G(I) +A(K,I)*R(K)^(K7-1)
2810 NEXT K
2820 G1 = G1 + G(I)*G(I)
2830 NEXT I
2840 G1 = SQR(G1)
2850 RETURN
2860 REM *************************************************************
2870 REM *   PENAMBAHAN PARAMETER  MARQUARDT  UNTUK MATRIK NORMAL     *
2871 REM *************************************************************
2880 L=0 :PRINT "          PARAMETER MARQUARDT ( V )          =";
2890 PRINT USING "####.###^^^^";V
2900 FOR J =  1 TO N
2910 FOR I = 1 TO N
```

```
2920 IF I<J THEN  GOTO  2950
2930 L = L + 1
2940 IF I=J THEN H(L) = H(L) + V*D(J)
2950 NEXT I
2960 NEXT J
2970 RETURN
2980 REM ****************************************************************
2990 REM *            TAMPILKAN SAMPEL  DATA DARI BARIS 400               *
2991 REM ****************************************************************
3000 PRINT " I      INDEPENDEN        DEPENDEN "
3010 K = 0
3020 FOR I =  1 TO M7
3030 PRINT  I; :PRINT USING "######.######";S(I,1);S(I,2)
3040 IF I<21 OR K=1 THEN GOTO 3070
3050 PRINT " TEKAN SEBARANG KUNCI UNTUK KONTINU "; : INPUT S4$
3060 K= 1 : REM - TANPA PAUSE UNTUK BERHENTI DARI DISPLAY
3070 NEXT I
3080 RETURN
3090 REM ****************************************************************
3100 REM *   LDLT FAKTORISASI UNTUK MATRIK DALAM SITU DALAM VEKTOR H   *
3104 REM *   N5 = 1 TIDAK DEFINIT POSITIF ATAU DET = D1 < D-6          *
3105 REM ****************************************************************
3110 K5=1 : N5=1 : D1 = 1
3120 FOR I = 2 TO N
3130 IF H(K5)>0  THEN GOTO 3150
3140 GOTO 3340
3150 Z = H(K5) : D1 = D1*H(K5)
3160 K5 = K5 +1
3170 I1 = K5
3180 FOR J = I TO N
3190 Z5 = H(K5)
3200 H(K5) = H(K5)/Z
3210 J5 = K5
3220 I5 = I1
3230 FOR K=I TO J
3240 J5 = J5 +N+1-K
3250 H(J5)=H(J5)-H(I5)*Z5
3260 I5=I5+1
3270 NEXT K
3280 K5=K5+1
3290 NEXT J
3300 NEXT I
3310 D1= D1*H(K5) : IF D1>.0000000001#  THEN N5=0
3320 IF N5=1 THEN GOTO 3340
3330 RETURN
3340 PRINT "HESSIAN TIDAK DEFINIT POSITIF ATAU DETERMINAN TERLALU KECIL";
3350 PRINT USING "##.####^^^^";D1
3360 RETURN
3370 REM ****************************************************************
3380 REM *        SOLUSI E = INV(H)E UNTUK LANGKAH PENYELIDIKAN        *
3381 REM ****************************************************************
3390 FOR I = 2 TO N
3400 I4 = I
3410 V5 = E(I)
3420 FOR J = 1 TO I-1
3430 V5= V5  - H(I4)*E(J)
3440 I4 = I4+ N -J
```

```
3450 NEXT J
3460 E(I) = V5
3470 NEXT I
3480 E(N)=E(N)/H(I4)
3490 FOR K = 2 TO N
3500 I = N+1-K
3510 I1 = I4 - K
3520 V5 = E(I)/H(I1)
3530 I4 = I1
3540 FOR J = I+1 TO N
3550 I1 = I1 + 1
3560 V5 = V5 -H(I1)*E(J)
3570 NEXT J
3580 E(I) = V5
3590 NEXT K
3600 RETURN
5000 REM ****************************************************************
5010 REM *        RESIDU UNTUK APROKSIMASI FUNGSI ROSENBROCK BANANA      *
5011 REM *                        M = JUMLAH SAMPEL                       *
5012 REM ****************************************************************
5020 M = 2
5030 L1 = 0
5040 R(1)= 10*(X(2)-X(1)*X(1))
5050 R(2) = 1 - X(1)
5060 RETURN
7000 REM ****************************************************************
7005 REM *             JAKOBIAN UNTUK FUNGSI ROSENBROCK BANANA            *
7010 REM ****************************************************************
7020 A(1,1)= -20*X(1)
7030 A(2,1)=-1
7040 A(1,2)= 10
7050 A(2,2) = 0
7060 RETURN
7070 END
```