Workshop on Differential Equations and its Applications
15-17 September 2004
Istanbul Technical University

Gradient Flow Method for Nonlinear Least Squares Minimization

Neculai Andrei
Research Institute for Informatics
Averescu Avenue, Bucharest 1, Romania
E-mail: nandrei@ici.ro

The problem:

$$min\,\Phi(x)$$
$$\Phi(x) = \frac{1}{2}\|F(x)\|^2,$$

where:

$$F(x) = [f_1(x), \ldots, f_m(x)]: R^n \to R^m$$

is twice continuous differentiable.
For practical situations $m \ge n$.

Approaches:

1) ***Gauss-Newton.***

$$x_{k+1} = x_k + d_k,$$

where:

$$\left(\nabla F(x_k)^T \nabla F(x_k)\right) d_k = -\nabla F(x_k)^T F(x_k).$$

2) ***Levenberg-Marquardt.***

$$\left(\nabla F(x_k)^T \nabla F(x_k) + \mu_k I\right) d_k = -\nabla F(x_k)^T F(x_k),$$

where
$\mu_k > 0$ controls both the magnitude and direction of d_k.

3) ***Gradient Flow.***

The necessary condition for optimality of x^* is:

$$\nabla\Phi(x^*) = 0,$$

where

$$\nabla\Phi(x) = \nabla F(x)^T F(x).$$

In order to fulfill this optimality condition the following continuous gradient flow reformulation of the problem is considered:

> ***"Solve the ODE:***
>
> $$\frac{dx(t)}{dt} = -\nabla\Phi(x(t)),$$
>
> ***with the initial condition*** $\quad x(0) = x_0$. ***"***

Theorem 1.

Consider that x^* is a point satisfying $\nabla\Phi(x^*) = 0$ and $\nabla^2\Phi(x^*)$ is positive definite. If x_0 is sufficiently close to x^*, then $x(t)$, the solution of the above ODE with initial condition x_0, tends to x^* as t goes to ∞.

Proof.

The above ODE can be written as $\dot{x} = \Psi(x)$, where $\Psi(x) = -\nabla\Phi(x)$. x^* is an asymptotically stable point for the nonlinear differential equation $\dot{x} = \Psi(x)$ if $\Psi(x)$ *is continuously differentiable* and the linearized system

$$\dot{y} = \nabla\Psi(x^*)y, \quad y = x - x^*,$$

is *exponentially stable*, i.e. all eigenvalues of $\nabla\Psi(x^*)$ are strictly negative.

We have:

$$\frac{dx}{dt} \cong \Psi(x^*) + \nabla\Psi(x^*)(x - x^*) = -\left[\nabla\Phi(x^*) + \nabla^2\Phi(x^*)(x - x^*)\right]$$
$$= -\nabla^2\Phi(x^*)(x - x^*).$$

But $\nabla^2\Phi(x^*)$ is positive definite, therefore all its eigenvalues $\lambda_i > 0$ for all $i = 1, \ldots, n$. Hence $\lim_{t\to\infty} y(t) = 0$, or $x(t) \to x^*$ as $t \to \infty$. ■

Theorem 2.
Let $x(t)$ be the solution of the above ODE with initial condition x_0. For a fixed $t_0 \geq 0$ if $\nabla\Phi(x(t)) \neq 0$ for all $t > t_0$, then $\Phi(x(t))$ is strictly decreasing with respect to t, for all $t > t_0$.

Proof.

$$\frac{d\Phi(x(t))}{dt} = \nabla\Phi(x(t))^T \frac{dx(t)}{dt} = -\nabla\Phi(x(t))^T \nabla\Phi(x(t)) = -\left\|\nabla\Phi(x(t))\right\|_2^2.$$

Since $\nabla\Phi(x(t)) \neq 0$ when $t > t_0$, it follows that $d\Phi(x(t))/dt < 0$, i.e. $\Phi(x(t))$ is strictly decresing with respect to $t > t_0$. ■

Conclusion:
Solving the unconstrained optimization problem $\quad min\,\Phi(x)$

has been reduced to that of integration of the ODE

$$\frac{dx(t)}{dt} = -\nabla\Phi(x(t)) \qquad \text{with} \qquad x(0) = x_0.$$

Discretization of the ODE
Let $0 = t_0 < t_1 < \cdots < t_k < \cdots$ be a sequence of time points for $t \geq t_0$.
Define $h_k = t_{k+1} - t_k$ the sequence of time distances.
Consider the following time-steeping discretization of the above ODE:

$$\frac{x_{k+1} - x_k}{h_k} = -\left[(1-\theta)\nabla\Phi(x_k) + \theta\nabla\Phi(x_{k+1})\right],$$

where $\theta \in [0,1]$ is a parameter.
When $\theta = 0$, then we have the ***explicit forward Euler's scheme***,
$\theta = 1$, then we have the ***implicit backward Euler's scheme***.
Omitting the higher order terms we get:

$$x_{k+1} = x_k - h_k\left[I + h_k\theta\nabla^2\Phi(x_k)\right]^{-1}\nabla\Phi(x_k),$$

for any $\theta \in [0,1]$.

Theorem 3.

Let $\{x_k\}$ be the sequence defined by

$$x_{k+1} = x_k - h_k\left[I + h_k\theta\nabla^2\Phi(x_k)\right]^{-1}\nabla\Phi(x_k)$$

and x^* a solution of the original problem, such that $\nabla^2\Phi(x^*)$ is positive definite. If the initial point x_0 is sufficiently close to x^*, then:

- ***(i)*** ***If $\theta \in [0,1]$ and $h_k > 0$ is sufficiently small, then x_k converges linearly to x^*.***
- ***(ii)*** ***If $\theta = 1$ and $h_k \to \infty$, then x_k converges quadratically to x^*.***

Proof (i)

After some algebra we get

$$\|e_{k+1}\| \leq \varphi(x_k, \xi_k, \theta, h_k)\|e_k\|,$$

where

$$\varphi(x_k, \xi_k, \theta, h_k) = \left\| I - h_k\left[I + h_k\theta\nabla^2\Phi(x_k)\right]^{-1}\nabla^2\Phi(\xi_k)\right\|,$$

$e_k = x_k - x^*$ and $\xi_k \in [x_k, x^*]$.

If $\varphi(x_k, \xi_k, \theta, h_k) < 1$, then e_k converges to zero linearly. Using continuity and the fact that x_0 is close to x^* we can write:

$$\varphi(x_k, \xi_k, \theta, h_k) \leq \left(1 - \frac{h_k\lambda_{\min}^k}{1 + h_k\theta\lambda_{\max}^k}\right) < 1,$$

where $\lambda_{\min}^k$ and $\lambda_{\max}^k$ are the minimum and the maximum eigenvalues of $\nabla^2\Phi(x_k)$, respectively.

Therefore $\lim_{k\to\infty} e_k = 0$ linearly, i.e. $x_k \to x^*$ linearly.

Proof (ii)

Considering $\theta = 1$ we get:

$$\frac{x_{k+1} - x_k}{h_k} = -\left[\nabla\Phi(x_k) + \nabla^2\Phi(x_k)\delta x_k\right],$$

where $\delta x_k = x_{k+1} - x_k$.

When $h_k \to \infty$ the above relation is reduced to:

$$\nabla\Phi(x_k) + \nabla^2\Phi(x_k)\delta x_k = 0,$$

which is the Newton method applied to $\nabla\Phi(x) = 0$. If x_0 is sufficiently close to x^*, then the convergence of the algorithm is quadratic.

Gradient Flow Algorithm (GFA)

Step 1. Consider x_0 , a parameter $\theta \in [0,1]$, a sequence of time step sizes $\{h_k\}$ and an $\varepsilon > 0$ sufficiently small. Set $k = 0$.

Step 2. Solve for d_k the system:

$$\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k) + \sum_{i=1}^{m} f_i(x_k)\nabla^2 f_i(x_k) \right)\right] d_k = -h_k \nabla F(x_k)^T F(x_k)$$

Step 3. Update the variables: $x_{k+1} = x_k + d_k$.

Step 4. Test for continuation of iterations. If $\|F(x_{k+1})\| \le \varepsilon$, STOP, otherwise set $k = k+1$ and go to step 2. ♦

To implement GFA algorithm we have

1) to compute the gradient and Hessian of the residual functions f_i ,
2) to select the value of θ and h_k ,
3) to solve a system of linear algebraic equations.

The *most difficult* is the task 1 above: to compute $\nabla f_i(x_k)$ and $\nabla^2 f_i(x_k)$.

According to theorem 3 above we see that if:

1) $f_i(x)$ are convex and positive for all $i = 1,\ldots,m$,
2) $rank(\nabla F(x)) = n$,
3) $\theta = 1$,
4) $h_k \to \infty$,

then the GFA is quadratically convergent to x^* .

Gradient Flow Algorithm with Scalar Approximation of Hessian

The ideea is to consider a scalar approximation of the Hessian matrices $\nabla^2 f_i(x_k)$ of residual functions $f_i(x)$, $i=1,\ldots,m$ at point x_k.

In the current point x_k the following approximation of the Hessian $\nabla^2\Phi(x_k)$ can be considered:

$$\nabla F(x_k)^T \nabla F(x_k) + \delta_k I ,$$

where the scalar δ_k can have the following values:

a) $\delta_k = \sum_{i=1}^{m} f_i(x_k)\gamma_i^k$ where $\gamma_i^k = \frac{2}{d_k^T d_k}\left[f_i(x_{k+1}) - f_i(x_k) - \nabla f_i(x_k)^T d_k \right].$

γ_i^k is a scalar approximation of the Hessian matrix of $f_i(x)$ in x_k.

b) $\delta_k = \sum_{i=1}^{m} f_i(x_k)^2 \left(\gamma_i^k\right)^2$.

c) *Procedure* δ *:*
Set $\delta_k = 0$.
For $i=1,\ldots,m$, do:
Set $p = f_i(x_k)$, $q = \gamma_i^k$.
If $p<0$, then $p = f_i(x_k)^2$.
If $q<0$, then $q = \left(\gamma_i^k\right)^2$.
Set $\delta_k = \delta_k + pq$.
End For.

d) $\delta_k = \sum_{i=1}^{m} f_i(x_k)^2$. $\nabla^2 f_i(x_k) = f_i(x_k) I$ (Dennis and Schnabel , [1983])

Modified Gradient Flow Algorithm (MGFA)

Step 1. Consider x_0, a parameter $\theta \in [0,1]$, a sequence of time step sizes $\{h_k\}$ and an $\varepsilon > 0$ sufficiently small. Compute $F(x_0)$, $\nabla F(x_0)$ and $\delta_0 = \|F(x_0)\|$. Set $k = 0$.

Step 2. Solve for d_k the system:

$$\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k) + \delta_k I\right)\right] d_k = -h_k \nabla F(x_k)^T F(x_k)$$

Step 3. Update the variables: $x_{k+1} = x_k + d_k$.

Step 4. Test for continuation of iterations. If $\|F(x_{k+1})\| \le \varepsilon$, STOP, otherwise set $k = k+1$ and go to step 5.

Step 5. Compute δ_k using one of the procedures: a), b), c) or d) and go to step 2. ♦

Theorem 4.

Let $\{x_k\}$ ***be the sequence defined by:***

$$x_{k+1} = x_k - h_k\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k) + \delta_k I\right)\right]^{-1} \nabla F(x_k)^T F(x_k)$$

and x^* ***a solution of the problem such that:***

(a) $F(x)$ ***is twice continuous differentiable,***
(b) $\nabla F(x)$ ***is Lipschitz continuous, and***
(c) $rank \nabla F(x^*) = n$.

If the initial point x_0 ***is sufficiently close to*** x^****, then:***

(i) ***If*** $\theta \in [0,1]$ ***and*** $h_k > 0$ ***is sufficiently small, then*** x_k ***converges linearly to*** x^****.***

(ii) ***If*** $\theta = 1$ ***and*** $h_k \to \infty$***, then*** x_k ***converges quadratically to*** x^****.***

Proof (i)

After some algebra we get:

$$\|e_{k+1}\| \le \varphi(x_k, \xi_k, \theta, \rho_k)\|e_k\|$$

where

$$\varphi(x_k, \xi_k, \theta, \rho_k) = \left\|I - \rho_k\left[I + \rho_k\theta\nabla F(x_k)^T \nabla F(x_k)\right]^{-1} \nabla F(x_k)^T \nabla F(\xi_k)\right\|,$$

$$\rho_k = \frac{h_k}{1 + h_k\theta\delta_k}.$$

When x_k is sufficiently close to x^* and if $\theta \in [0,1]$, by continuity it follows that

$$\varphi(x_k, \xi_k, \theta, \rho_k) \leq \left(1 - \frac{\rho_k \lambda_{\min}^k}{1 + \rho_k \theta \lambda_{\max}^k}\right),$$

where $\lambda_{\min}^k$ and $\lambda_{\max}^k$ are the minimum and the maximum eigenvalues of $\nabla F(x_k)^T \nabla F(x_k)$, respectively.
But

$$1 - \frac{\rho_k \lambda_{\min}^k}{1 + \rho_k \theta \lambda_{\max}^k} = 1 - \frac{h_k \lambda_{\min}^k}{1 + h_k \theta (\delta_k + \lambda_{\max}^k)} < 1.$$

Therefore

$$\lim_{k \to \infty} e_k = 0$$

linearly, i.e. x_k converges to x^* linearly.

Proof (ii)
Considering $\theta = 1$ we get:

$$\frac{x_{k+1} - x_k}{\rho_k} = -\nabla F(x_k)^T \left[F(x_k) + \nabla F(x_k)(x_{k+1} - x_k) \right].$$

But

$$\lim_{h_k \to \infty} \frac{h_k}{1 + h_k \delta_k} = \frac{1}{\delta_k}$$

and using, for example

$$\delta_k = \sum_{i=1}^{m} f_i(x_k)^2 \left(\gamma_i^k\right)^2$$

we see that $\lim\limits_{k \to \infty} \delta_k = 0$, i.e. $\lim\limits_{k \to \infty} \rho_k = \infty$.
Having in view that ∇F is of full column rank, we get:

$$\nabla F(x_k)(x_{k+1} - x_k) + F(x_k) = 0,$$

which is the Newton method applied to $F(x) = 0$. ■

Complexity of the algorithm

With $\theta = 1$ we can write:

$$\|e_{k+1}\| \le p_{k+1}\|e_0\|,$$

where

$$p_{k+1} = \prod_{i=0}^{k}\left(1 - \frac{h_i \lambda_{\min}^i}{1 + h_i(\delta_i + \lambda_{\max}^i)}\right).$$

But $\nabla F(x_i)^T \nabla F(x_i)$ is a positive definite matrix, therefore for all $i = 0, \ldots, k$,

$$0 < 1 - \frac{h_i \lambda_{\min}^i}{1 + h_i(\delta_i + \lambda_{\max}^i)} < 1.$$

Therefore, p_k is a decreasing sequence in $(0,1)$, i.e. p_k is convergent to zero.

Let

$$a_i = \frac{h_i \lambda_{\min}^i}{1 + h_i(\delta_i + \lambda_{\max}^i)} \qquad \text{and} \qquad a_j = \min\{a_i : 0 \le i \le k\}.$$

Then

$$p_{k+1} = \prod_{i=0}^{k}(1 - a_i) \le (1 - a_j)^{k+1}$$

i.e.

$$\|e_{k+1}\| \le p_{k+1}\|e_0\| \le (1 - a_j)^{k+1}\|e_0\|.$$

Therefore the ***number of iterations*** to get $\|e_{k+1}\| = \|x_{k+1} - x^*\| \le \varepsilon$, starting from x_0 is bounded by:

$$\frac{\log(\varepsilon) - \log(\|e_0\|)}{\log(1 - a_j)} - 1.$$

Levenberg-Marquardt algorithm

Considering $\delta_k = 0$ in MGF algorithm we get ***another*** algorithm which is very close to that of Levenberg and Marquardt:

$$x_{k+1} = x_k - h_k\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k)\right)\right]^{-1} \nabla F(x_k)^T F(x_k),$$

for which, like in theorem 4 above, for $\theta = 1$, we can prove that

$$\|e_{k+1}\| \le \bar{p}_{k+1}\|e_0\|,$$

where

$$\bar{p}_{k+1} = \prod_{i=0}^{k}\left(1 - \frac{h_i\lambda_{\min}^i}{1 + h_i\lambda_{\max}^i)}\right).$$

Theorem 5.

In the family of algorithms given by

$$x_{k+1} = x_k - h_k\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k) + \delta_k I\right)\right]^{-1} \nabla F(x_k)^T F(x_k),$$

the Levenberg-Marquardt algorithm, which correspond to $\theta = 1$ ***and*** $\delta_k = 0$***, is the best one.***

Proof.

In conditions of the theorem and having in view that $h_k > 0$ we can write:

$$\left[\frac{1}{h_k} I + \nabla F(x_k)^T \nabla F(x_k)\right] d_k = -\nabla F(x_k)^T F(x_k),$$

which is the Levenberg-Marquardt algorithm with $\mu_k = 1/h_k$.

Now, since $\nabla F(x_i)^T \nabla F(x_i)$ is positive definite and $\delta_i \ge 0$, it follows that for all $i = 0,1,\ldots,k$,

$$\frac{h_i\lambda_{\min}^i}{1 + h_i(\delta_i + \lambda_{\max}^i)} \le \frac{h_i\lambda_{\min}^i}{1 + h_i\lambda_{\max}^i}$$

i.e. $p_{k+1} \ge \bar{p}_{k+1}$. Therefore, with $\delta_k = 0$ and $\theta = 1$ the converegence of the algorithm is more rapid. ■

Numerical Example E1

$$f_1(x) = x_1^2 - 1,$$
$$f_i(x) = (x_{i-1} + x_i)^2 - i, \quad i = 2, \ldots, n.$$

Considering $\theta = 1$, $\varepsilon = 10^{-7}$, $x_0 = [1, \ldots, 1]$, the *number of iterations* are as follows:

Table 1a ($\delta_k = \sum_{i=1}^{m} f_i(x_k)^2 \left(\gamma_i^k\right)^2$)

n	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\lVert F(x_k)\rVert^2$
100	176	43	28	25	25	613
150	274	57	34	30	28	1600
200	378	71	38	34	32	3152

Table 1b (δ_k given by procedure δ)

n	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\lVert F(x_k)\rVert^2$
100	246	114	99	96	95	681
150	409	192	169	165	164	1732
200	586	279	247	242	241	3357

Table 1c ($\delta_k = \sum_{i=1}^{m} f_i(x_k)^2$)

n	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\lVert F(x_k)\rVert^2$
100	746	614	599	597	596	1179
150	1824	1607	1584	1581	1580	3145
200	3473	3166	3134	3130	3129	6242

Table 1d ($\delta_k = 0$)

n	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\lVert F(x_k)\rVert^2$
100	155	23	8	6	6	596
150	249	32	9	7	7	1580
200	350	42	11	7	7	3129

Numerical Example E2 ***(Circuit Design Problem)***

$$f_k(x) = (1 - x_1 x_2) x_3 \left\{ \exp\left[x_5 (g_{1k} - g_{3k} x_7 10^{-3} - g_{5k} x_8 10^{-3}) \right] - 1 \right\} - g_{5k} + g_{4k} x_2 \ ,$$
$$k = 1, \ldots, 4 \ ,$$

$$f_{4+k}(x) = (1 - x_1 x_2) x_4 \left\{ \exp\left[x_6 (g_{1k} - g_{2k} - g_{3k} x_7 10^{-3} - g_{4k} x_9 10^{-3}) \right] - 1 \right\} - g_{5k} x_1 + g_{4k}$$
$$k = 1, \ldots, 4 \ ,$$

$$f_9(x) = x_1 x_3 - x_2 x_4 \ ,$$

where

$$g = \begin{bmatrix} 0.4850 & 0.7520 & 0.8690 & 0.9820 \\ 0.3690 & 1.2540 & 0.7030 & 1.4550 \\ 5.2095 & 10.0677 & 22.9274 & 20.2153 \\ 23.3037 & 101.7790 & 111.4610 & 191.2670 \\ 28.5132 & 111.8467 & 134.3884 & 211.4823 \end{bmatrix}$$

The following initial point have been considered:

x_0^1	x_0^2	x_0^3	x_0^4
0.7	0.65	0.75	0.75
0.5	0.45	0.45	0.45
0.9	0.8	0.9	0.9
1.9	1.8	1.77	1.77
8.1	8.5	8.5	8.9
8.1	8.5	7.5	7.9
5.9	5.9	5.5	5.5
1	1.1	1.25	1.35
1.9	1.5	1.88	1.88

Considering $\theta = 1$, $\varepsilon = 10^{-7}$, the *number of iterations* are as follows:

Table 2a $(\delta_k = \sum_{i=1}^{m} f_i(x_k)^2 (\gamma_i^k)^2)$

	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\|F(x_k)\|^2$
x_0^1	142	50	40	38	38	27
x_0^2	173	60	47	45	45	56
x_0^3	256	146	133	131	131	132
x_0^4	600	500	489	487	487	488

Table 2b (δ_k given by procedure δ)

	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\|F(x_k)\|^2$
x_0^1	123	32	22	20	20	21
x_0^2	151	39	26	24	24	26
x_0^3	218	108	96	94	94	94
x_0^4	497	397	386	384	384	385

Table 2c $(\delta_k = \sum_{i=1}^{m} f_i(x_k)^2)$

	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\|F(x_k)\|^2$
x_0^1	113	22	12	10	10	11
x_0^2	140	27	15	12	12	14
x_0^3	135	25	13	11	11	12
x_0^4	124	24	14	12	11	13

Table 2d $(\delta_k = 0)$

	$h_k = 10$	$h_k = 10^2$	$h_k = 10^3$	$h_k = 10^4$	$h_k = 10^5$	$h_k = 1/\|F(x_k)\|^2$
x_0^1	108	10	6	4	4	10
x_0^2	132	16	7	5	4	12
x_0^3	129	19	6	5	5	11
x_0^4	46	15	6	5	5	11

Conclusion

The Problem:

$$\boxed{min\,\Phi(x)}$$

where:

$$\Phi(x) = \frac{1}{2}\|F(x)\|^2,$$

$$F(x) = [f_1(x), \ldots, f_m(x)] : R^n \to R^m$$

The Algorithm:

$$\boxed{\begin{array}{l} x_0 \ \text{given}, \\ x_{k+1} = x_k - h_k\left[I + h_k\theta\left(\nabla F(x_k)^T \nabla F(x_k) + \delta_k I\right)\right]^{-1} \nabla F(x_k)^T F(x_k), \\ k = 0,1,\ldots \end{array}}$$

The best results (Quadratic Convergence) are obtained for:

$\theta = 1$,
$h_k \to \infty$,
$\delta_k = 0$.

Advantages:
Very easy to implement,
Quadratic convergence,
There is no need to evaluate the Hessians $\nabla^2 f_i(x_k)$ of the residuals,
There is no need to do a linear search along the iterations.

Disadvantages:
A system of linear algebraic equations must be solved at each iteration.