ON THE BEHAVIOR OF NEWTON ITERATION

C. C. TSAI

ABSTRACT. We study the multivariate Newton method for the case of non-simple root. We give a criterion for it to converge, in which case the convergence rate will be linear. We then give a modified method to achieve an exponential rate of convergence. In contrast to this, we present also a number of examples to demonstrate some critical phenomenons of the Newton iteration when the criterion does not hold, and discuss possible ways to resolve the problem.

1. INTRODUCTION

The **Newton iteration** provides an algorithm to approximate the roots of smooth maps $f : R^n \to R^n$ from Euclidean n-space to itself. Namely

$$x \mapsto N(x) := x - (Df|_x)^{-1} f(x).$$

For the one-variable case it is well known that the Newton iteration always converges locally at any root, unless the derivative of all order vanish there. For multivariate case, the method is efficient if the root a is a simple root, i.e. the Jacobian of the map $\det Df|_a$ at a is non-zero. If $\det Df|_a = 0$, the Newton iteration is undefined at a, and therefore difficult for us to analyze its local behavior.

In section 2, we show that if the root is a zero of homogeneous order k, then the Newton iteration behaves like a *contraction mapping* of dissipation factor $(k-1)/k$. Furthermore, if f is dominated by its lowest homogeneous part in the sense that a *non-degeneracy condition* holds, then N behaves like the strictly homogeneous case (c.f. Theorem 2.1). In this case, we can even modify the Newton iteration to make the dissipation factor approximates 0, and therefore achieve an exponential rate of convergence (c.f. Theorem 2.3). The non-degeneracy holds trivially for one variable analytic case.

In section 3 we consider several examples, where the non-degeneracy fails, to demonstrate the *wild behavior* of Newton iteration. It may blows up along certain dangerous directions which are unavoidable. But in the (generalized) *homogeneous case*, a notion to be defined in section 3.1, we can still conclude that the Newton iteration is good at almost every direction, and all direction in certain case (c.f. Theorem 3.1).

For other situation, we separate them into two category: *non-homogeneous case* and *implicitly non-homogeneous case*. We begin the exploration of *non-homogeneous case* in section 3.2, where we show that the Newton iteration in general *won't be a contraction*, but it may *still converge*. In section 3.3 we give a non-convergent example of a smooth but non-analytic map which has a zero of infinite order.

In section 3.4, we give a non-homogeneous example randomly generated by computer, in which the Jacobian *never vanishes outside the root*, but the Newton iteration could *still diverge*. In section 3.5, we give another non-homogeneous example, in which the Newton iteration almost never converge to the root. Finally in section 3.6, *implicitly non-homogeneous case* is considered. We present two examples and show its very similarity and connection with the non-homogeneous cases.

Lastly, in section 4, we discuss the difficulties we have met in the non-homogeneous case. Some possible approaches are proposed, and we briefly discuss the obstructions one has to overcome when seeking for such a method.

There is an appendix section which contains the computer code in C++ which gives to all the examples discussed in section 3.

2. Criteria for convergence

2.1. **Simple roots.** Let us begin with a smooth map $f : R^n \to R^n$, the **Newton iteration** is defined to be the map $N(x) = x - D(x)$, where $D(x) = (Df|_x)^{-1} f(x)$. This arose from our interest in approaching the root of f. If a is a root of f where $\det Df|_a \neq 0$, namely a is a simple root, then a is a fixed point of N. In fact

$$DN|_x = I_n - D((Df|_x)^{-1})f(x) - (Df|_x)^{-1}Df|_x = -D((Df|_x)^{-1})f(x).$$

Hence $DN|_a = 0$ and N is a contraction mapping in a neighborhood of a. Moreover the Newton iteration converges exponentially there. Thus we only focus on the convergence issue at non-simple roots.

2.2. **The strictly homogeneous case.** Suppose now the origin 0 is a root of f. We are interested in the behavior of N near 0. Let $f = (f_1, f_2, \dots, f_n)$. Suppose that f_i are algebraic and homogeneous of the same degree k. Then we have Euler's relation $Df|_x x = kf(x)$ (by differentiating the equation $f(\lambda x) = \lambda^k f(x)$ at $\lambda = 1$). Hence

$$Df|_x \frac{x}{k} = f(x), \quad D(x) = \frac{x}{k}, \quad N(x) = \frac{k-1}{k} x$$

and the Newton iteration converges to 0 with a fixed linear rate.

We remark here that the Newton iteration is invariant under invertible linear transformation, both on the domain and on the range space.

2.3. **A non-degeneracy condition.** Now suppose that each f_i has k zeroes at the origin, i.e. $f_i = P_i + R_i$, where P_i is a polynomial of degree k and the first derivative of the remainder is bounded

$$DR_i \leq M|x|^k$$

in a neighborhood of the origin. We further assume that $|P_i| \le |x|^k$ (note that we're free to scale f_i) and at some point x_0 in the neighborhood the non-degeneracy condition holds: There is a constant C such that

$$|\,(DP|_{x_0})^{-1}\,| \le C|x_0|^{-k+1},$$

where $P = (P_1, \dots, P_n)$. Then we have

Theorem 2.1. *Under the non-degeneracy assumption on f at x_0, we have*

$$D(x_0) = x_0/k + \epsilon(x_0) \quad \textit{with} \quad |\epsilon(x_0)| \le BC^2|x_0|^2,$$

where the bound B depends only on n and M. In particular if f is non-degenerate in a neighborhood of a root, the the Newton iteration converges locally with rate being approximately $1 - 1/k$.

Proof. Notice that $DR_i \le M|x|^k$ gives $R_i \le M|x|^{k+1}$ in a neighborhood of the origin. We shall estimate the effect of the Newton iteration at x_0.

Since by assumption we know $(DP)^{-1}P$ very well ($= x/k$), it suffices to estimate $|(DP)^{-1}P - (DP)^{-1}f|$ and $|(DP)^{-1}f - (Df)^{-1}f|$.

For the first we have

$$|(DP)^{-1}P - (DP)^{-1}f| \le |(DP)^{-1}||P - f| \le CMn^{1/2}|x_0|^2.$$

For the second we use the fact that if x is a solution for $Ax = b$ and $x + \epsilon x$ is a solution for $(A + \delta A)(x + \epsilon x) = b$, then

$$\begin{aligned} & (\delta A)x + A(\epsilon x) + (\delta A)\epsilon x = 0 \\ \Longrightarrow \quad & A(I_n + A^{-1}\delta A)\epsilon x = -(\delta A)A^{-1}b \\ \Longrightarrow \quad & |\epsilon x| \le 2(|b| \cdot |\delta A| \cdot |A|^{-2}), \text{ provided } |A^{-1}\delta A| < \frac{1}{2}. \end{aligned}$$

By applying this to $A = Df|_{x_0}$, $\delta A = DR|_{x_0}$ and $b = f(x_0)$, we have

$$|A^{-1}\delta A| \le CMn^{1/2}|x| < \frac{1}{2}$$

if $|x| < \frac{1}{2CMn^{1/2}}$. And $|b| \le |P| + |R| \le n^{1/2}|x|^k + n^{1/2}M|x|^{k+1}$. Hence

$$|(DP)^{-1}f - (Df)^{-1}f| \le nMC^2|x_0|^2 + nM^2C^2|x_0|^3.$$

From

$$D(x_0) = (Df|_{x_0})^{-1}f = (DP|_{x_0})^{-1}P + ((DP|_{x_0})^{-1}P - (Df|_{x_0})^{-1}f)$$

and the above two estimates (we may assume that $|x_0| < 1$), we conclude that

$$|D(x_0) - x_0/k| < nCM(1 + CM + CM^2)|x_0|^2.$$

$\square$

Remark 2.2. *Theorem 2.1 and the non-degenerate condition can be regarded as the simplest generalization of the one variable case with multiple roots.*

2.4. **A modified method.** Suppose that for all x satisfying $|\frac{x}{|x|} - \frac{x_0}{|x_0|}| < \delta$ we have the non-degeneracy $|DP|_x|^{-1} \leq C|x|^{-k+1}$.

Let $A = BC^2(\delta^{-1}+1)+1$. Then we begin with

$$|x_0| \leq \frac{A^{-1}}{16k},$$

and let

$$x_1 = x_0 - (1-2^{-3})kD(x_0),$$

$$x_2 = x_1 - (1-2^{-4})kD(x_1),$$

$$\vdots$$

$$x_{n+1} = N_n(x_n) = x_n - (1-2^{-(2^n+2)})kD(x_n),$$

Note that we can compute multiples of $D(x)$, since $D(x)$ is invariant under translation of coordinates. Inductively we have

Theorem 2.3. *Begin with the condition assumed as above, we have*

$$|x_n| \leq \frac{A^{-1}}{2^{C_n}k}, \qquad \left|\frac{x_0}{|x_0|} - \frac{x_n}{|x_n|}\right| < \delta,$$

where $C_n = 2^n + n + 3$ and the x_n's are given by the iteration

$$x_{n+1} = x_n - (1-2^{-(2^n+2)})kD(x_n).$$

Proof. We claim

$$|x_n| \leq \frac{A^{-1}}{2^{C_n}k}, \qquad \left|\frac{x_n}{|x_n|} - \frac{x_{n+1}}{|x_{n+1}|}\right| < 2^{-n}\delta.$$

Indeed, suppose it holds for $n-1$. Then

$$x_{n+1} = 2^{-(2^n+2)}x_n + (1-2^{-(2^n+2)})(x_n - kD(x_n))$$

which, by theorem 2.1, implies that

$$(*) \qquad |x_{n+1} - 2^{-(2^n+2)}x_n| \leq BC^2k|x_n|^2 \leq \frac{2^{-C_n}}{\delta^{-1}+1}|x_n|.$$

On one hand, this shows that

$$|x_{n+1}| \leq \left(2^{-(2^n+2)} + \frac{2^{-C_n}}{\delta^{-1}+1}\right)|x_n| \leq 2^{-(2^n+1)}|x_n| \leq \frac{A^{-1}}{2^{C_{n+1}}}.$$

On the other hand, letting $\bar{x}_n = 2^{-(2^n+2)}x_n$,

$$\left|\frac{x_n}{|x_n|} - \frac{x_{n+1}}{|x_{n+1}|}\right| = \left|\frac{\bar{x}_n}{|\bar{x}_n|} - \frac{x_{n+1}}{|x_{n+1}|}\right| \leq \left|\frac{\bar{x}_n}{|\bar{x}_n|} - \frac{x_{n+1}}{|\bar{x}_n|}\right| + \left|\frac{x_{n+1}}{|\bar{x}_n|} - \frac{x_{n+1}}{|x_{n+1}|}\right|.$$

The first term of the RHS is controlled by $(*)$. It is no greater than $\frac{2^{-(n+1)}}{\delta^{-1}+1}$. Using the estimation of $|x_{n+1}|$ together with $(*)$, and keep in mind that $|\frac{1}{a}-\frac{1}{b}| \leq \frac{|a-b|}{ab}$, we have

$$\left|\frac{x_{n+1}}{|\bar{x}_n|}-\frac{x_{n+1}}{|x_{n+1}|}\right| \leq \frac{2^{-C_n}}{\delta^{-1}+1}\cdot\frac{|x_{n+1}|\cdot|x_n|}{|\bar{x}_n|\cdot|x_{n+1}|} \leq \frac{2^{-(n+1)}}{\delta^{-1}+1} < 2^{-(n+1)}\delta.$$

Hence the sum of the two terms is smaller than $2^{-n}\delta$, as promised. □

This gives a method to converge to the root by exponential rate (as the rate of Newton method for usual simple root), provided that we know k. We may detect k by letting $|x_0| \ll A^{-1}$, and compare the ratio between $|x_0 - N(x_0)|$ and $|N(x_0) - N(N(x_0))|$. They should be about $\frac{k}{k-1}$, as can be controlled by the above estimate.

3. Examples and phenomenons

3.1. The necessity of the bound C. Here we give an example here to show that the assumption in Theorem 2.1 and Theorem 2.3 about the constant C is necessary. Consider

$$\begin{aligned} f_1 &= 2x^2 - 5xy + 2y^2 + x^2y^2, \\ f_2 &= xy. \end{aligned}$$

The Newton iteration is given by

$$N\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2x^2-2y^2}\begin{pmatrix} x^3y^2 \\ -x^2y^3 \end{pmatrix}$$

When $|x|$ is close to $|y|$. $N\binom{x}{y}$ will blow up; $N\binom{x}{y}$ become very big after the first step of iteration, and it will continue going toward infinity. Hence it will be impossible to approximate the root by the Newton iteration if we begin at a bad direction.

However, one sees that in Theorem 2.3 we only require that x_0 is smaller by some multiple of A^{-1}. Hence if we consider B fixed, C can be taken to be of the order $O(|x_0|^{-0.5})$. This shows that if the point we begin is close enough to the root, then almost every direction will leads us to the root. Indeed, we only asked for the assumption

$$|DP|_{x_0}|^{-1} \leq C|x_0|^{-(k-1)}$$

But $|\,(DP|_x)^{-1}\,| \leq (\det DP|_x)^{-1}\,|DP|_x|^{n-1}$, which should be of $-(k-1)$ degree in x, thus such C must exist for any fixed direction (i.e. on a line through the origin), unless we are on the hypersurface $\det DP = 0$ of codimension 1, or $\det DP$ is the zero polynomial.

When all the P_i's are of the same degree and $\det DP$ is not the zero polynomial, we call the case **homogeneous**. When all the P_i's are of the same degree but $\det DP$ *is* the zero polynomial, we call the case **implicitly non-homogeneous**, and **non-homogeneous** when some of the P_i's are not of the same degree.

We have seen the good behavior of homogeneous case. The rest of this chapter will deal with non-homogeneous cases, in which we postpone the explanation and discussion of implicit non-homogeneity to section 3.6.

We also remark here that if $\det DP$ is positive/negative definite (that is, non-vanishing except at the origin), then with some control of the kind $|DP|_x|^{n-1} \leq N|x|^{k-1}$ (recall that we're free to scale f and P), we may choose a large enough C for any direction, i.e. Theorem 2.1 and Theorem 2.3 holds for all x in a neighborhood of the origin. In summary we have

Theorem 3.1. *In homogeneous case, if* $\det DP$ *is positive/negative definite, then the Newton iteration is a contraction in a neighborhood of the root* x_0 *with the dissipation factor* $\to (k-1)/k$ *as* $x \to x_0$. *And we have a modified method to make the factor* $\to 0$ *and thus achieve an exponential rate of convergence.*

3.2. **A non-contraction example which "converges".** In section 2 it seems like things just work like a contraction; we have $N(x) \backsim \frac{k-1}{k}x$. That is, however, not the general case.

Recall that when each P_i are homogeneous of degree k. We have $DP|_x \frac{x}{k} = P(x)$. But if now each P_i are of degree k_i not all the same, then we have

$$(DP_i|_x)\,\frac{x}{k_i} = P_i(x).$$

But then we can have no good guess about $(DP|_x)^{-1}P(x)$. Consider

$$f_1 = x^2,$$
$$f_2 = x^3 + y^5.$$

The Newton iteration is given by

$$N\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x \\ \frac{x^3}{10y^3} + \frac{4}{5}y \end{pmatrix}.$$

This is not a contraction mapping. In fact, $N(x,y)$ does not even converge to $(0,0)$ as (x,y) approaches $(0,0)$; when x and y are of about equal magnitude, the term $x^3/10y^3$ is constant. However, one sees that under Newton iteration, x does decreases steadily, in a rate which is generally faster than y. Hence x will converges to zero, and y will follow x to the zero, in a relatively slow rate, making the term $x^3/10y^3$ not a threat.

To be precise, suppose at some step $|x|^3 < |y|^4$. Writing $\binom{x'}{y'} = N\binom{x}{y}$, we have

$$|x^3/10y^3| < |y|/10 \Rightarrow 0.7 < |y'|/|y| < 0.9$$

Hence we have two estimate: $|y'| < 0.9|y|$ and $|x'|^3 = \frac{1}{8}|x|^3 < (0.7)^4|y|^4 < |y'|^4$. In this case by induction we see that the Newton iteration converges. If otherwise $|y|^4 \leq |x|^3$ at all steps, then since x always decreases to zero, the Newton iteration also converges. Thus unless the denominator vanishes at some step, the Newton iteration will send any point toward the origin, the only root of this system.

It is in fact no surprise that we have a non-contraction convergence. The Newton iteration is totally decided by the linear structure of R^n, while when we talk about contraction, it has to do with metric. Hence the only chance for it to be a contraction is that the vector shrink by a factor $N(x) = \rho x$, with $\rho < 1$. And we have seen that it happens only for homogeneous cases.

3.3. **The order of zero = ∞.** We already know that when k is a fixed integer, the contraction factor (for homogeneous case) is $(k-1)/k$. Now we consider the following smooth real-valued function

$$f(x) = \phi(x)e^{\frac{1}{|x|}}, \quad \text{where} \quad \phi(x) = 100 - 3\sin^3 x\cos x - 5\sin x\cos^3 x.$$

And we define $f(0) = 0$. We have $\phi'(x) = 3 - 8\cos^4 x$ and the Newton iteration is given by

$$N(x) = x - f(x)/f'(x) = x + x^3\left(\frac{6 - 16\cos^4 x}{100 - 3\sin^3 x\cos x - 5\sin x\cos^3 x} - x\right)^{-1}.$$

The second term may be arbitrary large. Moreover it is more likely to be of the same sign with x rather than opposite sign. This shows that the finiteness of the order of zero is somehow necessary.

3.4. **Non-homogeneous example I.** Next we observe two non-homogeneous examples. Here is the first one:

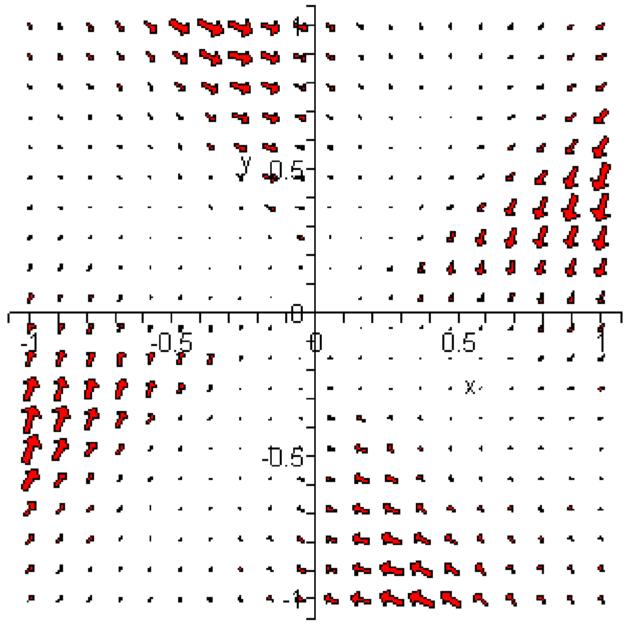
$$f_1 = 5x^2 + 3xy - 5y^2,$$
$$f_2 = 14x^4 - 7x^3y - 4xy^3 - 11y^4.$$

The Newton iteration is given by

$$N\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \dfrac{-161x^5 + 462x^4y - 235x^3y^2 - 220x^2y^3 - 93xy^4 - 110y^5}{-238x^4 + 602x^3y - 330x^2y^2 - 464xy^3 - 172y^4} \\ \dfrac{140x^5 - 147x^4y + 280x^3y^2 - 205x^2y^3 - 354xy^4 - 119y^5}{-238x^4 + 602x^3y - 330x^2y^2 - 464xy^3 - 172y^4} \end{pmatrix}.$$

Let $S_1 = \{(x,y) \mid y/x \in [-3.9, -3.3]\}$, $S_2 = \{(x,y) \mid x/y \in [2, 2.3]\}$. Then $N(S_1) \subset S_2$, $N(S_2) \subset S_1$. And if we begin with any $x \in S_1$ or S_2 then the Newton iteration will send it to infinity. The picture below show the direction of $-D(x) = N(x) - x$. When x is at the upper left region $\subset S_1$, $N(x)$ will be in the right upper region $\subset S_2$, then $N(N(x))$ will be in the lower right region $\subset S_1$, and so on. And it will be sent further and further (although pretty slowly) from the origin.

Furthermore, one notes that in this example, Df (that is, the denominator above) never vanishes except at the root, but the Newton iteration still



diverge. This shows that homogeneity is essential in the general convergence of Newton iteration.

3.5. **Non-homogeneous example II.** The second example is given by

$$f_1 = y - x^3 + x^{100}y^{100},$$
$$f_2 = y^2 - x^5 + x^{99}y^{99}.$$

If $X = \binom{x}{y}$ is close to zero and is not close to the x-axis (that is, the common tangent of f_1 and f_2) in the sense that x/y is bounded, then the Newton iteration will sends X far away along the x-axis. This may be explained by that both ∇f_1 and ∇f_2 are almost parallel to the y-axis, but the ratios

$$\left|\frac{\nabla f_1}{f_1}\right| \quad \text{and} \quad \left|\frac{\nabla f_2}{f_2}\right| \quad \text{are distinct.}$$

Indeed, recalled that in Newton iteration, the modification $D(x)$ is to satisfies the property

$$Df|_x D(x) = f(x), \text{which becomes} \begin{pmatrix} 0+\epsilon & 1+\epsilon \\ 0+\epsilon & 2y+\epsilon \end{pmatrix} D(x) = \begin{pmatrix} y+\epsilon \\ y^2+\epsilon \end{pmatrix}$$

where the $+\epsilon$ roughly means "smaller terms". If we see the above equation as two linear equations, it means that $D(x)$ has to be the intersection of two lines almost parallel to the x-axis. At the origin, the distance of the two lines looks like $\frac{y}{2}$ (consider setting all $\epsilon = 0$). Hence the solution $D(x)$ will be driven far away along the x-axis, depending on the behavior of the small terms and how small they are. They can be very small, and in particular in this example we have

$$D\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{y^2-2x^3y+x^5+98x^{99}y^{99}-99x^{102}y^{98}-98x^{100}y^{101}+100x^{105}y^{99}-x^{199}y^{198}}{-6x^2y+5x^4-99x^{98}y^{99}+200x^{99}y^{101}-297x^{101}y^{98}+500x^{104}y^{99}} \\ \frac{-3x^2y^2+5x^4y-2x^7-99x^{98}y^{100}+96x^{101}y^{99}+100x^{99}y^{102}-95x^{104}y^{100}+x^{198}y^{199}}{-6x^2y+5x^4-99x^{98}y^{99}+200x^{99}y^{101}-297x^{101}y^{98}+500x^{104}y^{99}} \end{pmatrix}.$$

When x, y are very small, we have

$$N\begin{pmatrix} x \\ y \end{pmatrix} \backsim \frac{1}{6x^2y}\begin{pmatrix} -y^2+4x^3y \\ 3x^2y^2 \end{pmatrix}.$$

After X has been sent far away along the x-axis by the first step, the second step will send it further (and not along the x-axis), it will then be sent toward infinity because of the higher order terms. If f_1 and f_2 are merely smooth, then anything can happen after the first step, e.g. it may converge to another root.

3.6. **About implicit non-homogeneity.** Recall that implicit non-homogeneity happens when the P_i's are of the same degree but $\det DP$ is the zero-polynomial. First we give an example.

$$f_1 = 5x^2 + 3xy - 5y^2 - 14x^4 + 7x^3y + 4xy^3 + 11y^4,$$
$$f_2 = 5x^2 + 3xy - 5y^2 + 14x^4 - 7x^3y - 4xy^3 - 11y^4.$$

One sees immediately that after a simple linear transformation, this is nothing but the example in 3.4. That's why we call it implicitly non-homogeneous. In this case, the P_i's are just all the same. And thus they are of the same degree, but $\det DP$ is definitely zero.

For $n = 2$, the implicit non-homogeneity is just non-homogeneity, hidden by a linear transformation. However, for $n = 3$ things becomes more complicated. Consider

$$f_1 = x^2,$$
$$f_2 = y^2,$$
$$f_3 = xy + x^2y^2 + z^8 - z^6.$$

In this case all the P_i's are independent of z, and thus $\det DP = 0$. The Newton iteration is given by

$$N(x, y, z) = \left(\frac{1}{2}x, \quad \frac{1}{2}y, \quad z - \frac{z^8 - z^6 - x^2y^2}{8z^7 - 6z^5}\right)$$

It has similar property with the example in 3.5. Begin with small x, y, z, not too close to the z-axis in the sense that $|z^2/xy|$ is bounded, The z-component becomes large after the first step. Later steps of Newton iterations then send it to the root $(0, 0, 1)$ or $(0, 0, -1)$; it almost never converges to the root $(0, 0, 0)$.

4. Discussion

All the results above may be generalized without change to the complex case. We have seen that things are not bad when f is homogeneous. When f happens to be (explicitly or implicitly) non-homogeneous, then the Newton iteration may break down. And when it happens that for two components f_i, f_j we have $P_i^a = P_j^b$ with $a \neq b$ (that is the general form for the example in 3.5; they have similar tangents but distinct degree), the root becomes almost impossible to be approached by the Newton iteration; roughly we have $\lim_{x\to 0} N(x) \longrightarrow \infty$ for almost every direction. To overcome this problem, one may pick a f_i and solve for $Df_i = 0$. However this may still break down. For example, if we pick Df_2 in

$$\begin{aligned} f_1 &= y - x^3, \\ f_2 &= y^2 - xy^3 + x^7. \end{aligned}$$

Then we see that Df_2 also has a root in the origin, but Newton iteration will not approach the root for the same reason as above. Another more general problem is

$$\begin{aligned} f_1 &= y - x^3, \\ f_2 &= y + y^2 - x^3 - xy^3 + x^7. \end{aligned}$$

This is the implicitly non-homogeneous version of the above. Since $Df_1, Df_2 \neq 0$ at the origin, we gain nothing by considering Df_1 or Df_2, unless we know how to "separate" $y^2 - xy^3 + x^7$ from f_1 and f_2. The point is that implicitly non-homogeneous cases seem to have all the troubles which happen in (explicitly) non-homogeneous case, but it might be harder to detect it.

Such separation problem seems to be an essential difficulty. If we can overcome it, then we can perform some transformation to make it better. For example, in the first example in this section, if we know to square $f_1 = f_1^2 = y^2 - 2x^3y + x^6$, and furthermore apply linear transformation $f_1 = f_1 - f_2$. it becomes

$$\begin{aligned} f_1 &= xy^3 - 2x^3y + x^6 - x^7, \\ f_2 &= y^2 - xy^3 + x^7. \end{aligned}$$

That is the case as in 3.4. Now, if again we know to square f_2, then it becomes homogeneous:

$$\begin{aligned} f_1 &= xy^3 - 2x^3y + x^6 - x^7, \\ f_2 &= y^4 - 2xy^5 + x^2y^6 + 2x^7y^2 - 2x^8y^3 + x^{14}. \end{aligned}$$

This is the case that we can control. However, since our purpose is to find the root, we cannot pretend that we know the root is at the origin, hence we cannot talk about the degree of f_i directly; one needs another method to "detect" the degree in the non-homogeneous case.

One should also note that, as shown in 3.6, implicit non-homogeneity can introduce greater difficulty for the separation. Recall the example

$$\begin{aligned} f_1 &= x^2, \\ f_2 &= y^2, \\ f_3 &= xy + x^2y^2 + z^8 - z^6, \end{aligned}$$

in which the implicit non-homogeneity cannot be resolved by linear transformation. We might need algebraic technique, e.g. take $\bar{f}_3 = (f_3 - f_1 f_2)^2 - f_1 f_2$.

5. Appendix — Program codes

The following is a program using standard C++ language. In the line "#define CASE 1", one may change the 1 to 2, 3, 4, 5, or 6. The resulting program corresponds to the example in 3.1, 3.2, 3.3, 3.4, or 3.5, or the second example in 3.6, respectively.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define CASE 1
#define N 210
double x, y, z;
struct poly{
   int co[N][N];
};
poly Dx(poly p){
   poly tmp;
   int i, j;
   for (i=0; i<N; i++){
      for (j=0; j<N; j++){
         if (i == N - 1)
            tmp.co[i][j] = 0;
         else
            tmp.co[i][j] = p.co[i+1][j]*(i + 1);
      }
   }
   return tmp;
}
poly Dy(poly p){
   poly tmp;
   int i, j;
   for (i = 0; i < N; i++){
      for (j = 0; j < N; j++){
         if (j == N - 1)
            tmp.co[i][j] = 0;
         else
```

```
            tmp.co[i][j] = p.co[i][j + 1]*(j + 1);
         }
      }
      return tmp;
   }
   double eval(poly p){
      double px[N],py[N];
      double sum = 0;
      int i, j;
      px[0] = py[0]= 1;
      for (i = 1; i < N; i++)
         px[i] = px[i-1]*x, py[i] = py[i - 1]*y;
      for (i = 0;i < N;i ++){
         for (j = 0; j ¡ N; j++){
            if (p.co[i][j]! = 0)
               sum+ = px[i]*py[j]*p.co[i][j];
         }
      }
      return sum;
   }
   poly f,g;
   void newton(){
      double fx = eval(Dx(f)), fy = eval(Dy(f)), gx = eval(Dx(g)), gy = eval(Dy(g));
      double det = fx*gy - fy*gx;
      double dx = gy*eval(f) - fy*eval(g), dy = fx*eval(g) - gx*eval(f);
      x = x - dx/det, y = y - dy/det;
   }
   int main(){
      int c, t, k, ch;
      double tx, ty;
      if (CASE == 1){
         f.co[2][0] = 2, f.co[1][1] = -5, f.co[0][2] = 2, f.co[2][2] = 1;
         g.co[1][1] = 1;
      }
      if (CASE == 2){
         f.co[2][0] = 1;
         g.co[3][0] = 1, g.co[0][5] = 1;
      }
      if (CASE == 3){
         scanf("%lf",&x);
         while(1){
            y=1/(x*x);
            for(int t=0;t¡100000;t++)
               x+=x*x*x/((3-8*pow(cos(y),4))/(100.0-1.5*pow(sin(y),3)*cos(y)-2.5*sin(y)*pow(cos(y),3))-
x);
            printf("%.12lf",x);
            system("pause");
         }
```

```
  }
  if (CASE == 4){
    f.co[2][0] = 5, f.co[1][1] = 3, f.co[0][2] = -5;
    g.co[4][0] = 14, g.co[3][1] = -7, g.co[2][2] = 0;
    g.co[1][3] = -4, g.co[0][4] = -11;
  }
  if (CASE == 5){
    f.co[0][1] = 1, f.co[3][0] = -1, f.co[100][100] = 1;
    g.co[0][2] = 1, g.co[5][0] = -1, g.co[99][99] = 1;
  }
  if (CASE == 6){
    scanf("%lf %lf %lf",&x,&y,&z);
    while(1){
      x/=2,y/=2,z-=(pow(z,8)-pow(z,6)-x*x*y*y)/(8*pow(z,7)-6*pow(z,5));
      printf("%.12lf %.12lf %.12lf",x,y,z);
      system("pause");
    }
  }
  scanf("%lf %lf",&x,&y);
  while(1){
    newton();
    printf("%.12lf %.12lf \ n", x, y);
    system("pause");
  }
  system("pause");
}
```

Department of Mathematics, National Taiwan University
E-mail address: b95201056@ntu.edu.tw