# Chapter 9 Newton iteration

## 9.1 Division with remainder using Newton iteration

For all $a, b$ with $b \neq 0$ in the Euclidean domain ($\mathbb{Z}$, $\mathbb{Q}[x]$), find unique $q, r$ such that $a = q \cdot b + r$.

1. classical division algorithm ($O(n^2)$)

2. division using Newton inversion($O(\mathsf{M}(n))$, M is a multiplication time)

Let $D$ be a ring(commutative, with 1) and $a, b \in D[x]$ two polynomials of degree $n \geq m$ respectively, $b$ is monic. Find $q, r \in D[x]$, $\deg r < \deg b$

$$a(x) = q(x) \cdot b(x) + r(x) \quad (*)$$

We define the Reversal of $a$ as $\text{rev}_k(a) = x^k a(1/x)$. If $a = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, then

$$\text{rev}_n(a) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n.$$

(*) becomes

$$\text{rev}_n(a) = \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) + x^{n-m+1}\text{rev}_{m-1}(r).$$

Therefore

$$\text{rev}_n(a) \equiv \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) \mod x^{n-m+1}.$$

Since $b$ is monic, $\mathsf{rev}_m(b)$ has constant coefficient 1 and thus is invertible modulo $x^{n-m+1}$,

$$\mathsf{rev}_{n-m}(q) \equiv \mathsf{rev}_n(a) \cdot \mathsf{rev}_m(b)^{-1} \mod x^{n-m+1}.$$

obtain

$$q = \mathsf{rev}_{n-m}(\mathsf{rev}_{n-m}(q)),\ r = a - q \cdot b.$$

See example 9.1.

For a given polynomial $f \in D[x]$, $f(0) = 1$ and $l \in \mathbb{N}$, find a polynomial $g \in D[x]$ satisfying

$$f \cdot g \equiv 1 \mod x^l$$

Define $\phi(g) = \frac{1}{g} - f$, the Newton iteration step is

$$g_{i+1} = g_i - \frac{\phi(g_i)}{\phi'(g_i)} = g_i - \frac{1/g_i - f}{-1/g_i^2} = 2g_i - fg_i^2$$

**Theorem 1.** *Let $D$ be a ring, $f, g_0, g_1, \ldots \in D[x]$, with $f(0) = 1, g_0 = 1$, and $g_{i+1} \equiv 2g_i - fg_i^2 \mod x^{2^{i+1}}$ for all $i$. Then $f \cdot g_i \equiv 1 \mod x^{2^{i+1}}$ for all $i \geq 0$.*

Proof: induction

$$1 - fg_{i+1} \equiv 1 - f(2g_i - fg_i^2) \equiv 1 - 2fg_i + f^2g_i^2 \equiv (1 - fg_i)^2 \equiv 0 \mod x^{2^{i+1}}$$

Continue of example 9.1.

## Algorithm 9.3 Inversion using Newton iteration

Input: $f \in D[x]$ with $f(0) = 1$, and $l \in \mathbb{N}$.

Output: $g \in D[x]$ satisfying $f \cdot g \equiv 1 \mod x^l$.

1. $g_0 \longleftarrow 1$, $r \longleftarrow \lceil \log l \rceil$
2. for $i = 1, \ldots, r$ do $g_i \longleftarrow (2g_{i-1} - f g_{i-1}^2)$ rem $x^{2^i}$
3. return $g_r$.

**Theorem 2.** *Algorithm 9.3 correctly computes the inverse of $f$ modulo $x^l$. If $l = 2^r$, then it uses at most $3M(l) + l \in O(M(l))$ arithmetic operations in $D$.*

## Algorithm 9.5 Fast division with remainder

Input: $a, b \in D[x]$ where $D$ is a ring and $b \neq 0$ is monic.

Output: $q, r \in D[x]$ such that $a = qb + r$ and $\deg r < \deg b$.

1. if $\deg a < \deg b$ then return $q = 0$ and $r = a$

2. $m \longleftarrow \deg a - \deg b$, compute $\mathrm{rev}_{\deg b}^{-1}(b) \mod x^{m+1}$

3. $q^* \longleftarrow \mathrm{rev}_{\deg a}(a) \cdot \mathrm{rev}_{\deg b}^{-1}(b)$ rem $x^{m+1}$

4. return $q = \mathrm{rev}_m(q^*)$ and $r = a - b \cdot q$

**Theorem 3.** *Let $D$ be a ring (commutative, with 1). Division with remainder of a polynomial $a \in D[x]$ of degree $n+m$ by a monic polynomial $b \in D[x]$ of degree $n$, $m \geq n$ can be done using $4M(m) + M(n) + O(n)$ ring operations.*

**Corollary 1.** *Let $D$ be a ring(commutative with 1) and $f \in D[x]$ monic of degree $n$. Then one multiplication in the residue class ring $D[x]/ < f >$ can be done using $6M(n) + O(n)$ arithmetic operations in $D$.*

**Theorem 4.** *For an integer $m \in \mathbb{N}$ of length $n$, one multiplication in the residue class ring $Z_m$ can be performed using $O(M(n))$ word operations.*

## Algorithm 9.10 $p$-adic inversion using Newton iteration

Input: $f, g_0 \in R$ with $f \cdot g_0 \equiv 1 \mod p$, where $R$ is a ring(commutative, with 1) and $p \in R$ arbitrary, and $l \in \mathbb{N}$.

Output: $g \in R$ satisfying $f \cdot g \equiv 1 \mod p^l$.

1. $r \longleftarrow \lceil \log l \rceil$
2. for $i = 1, \ldots, r$ compute $g_i \in R$ such that $g_i \equiv (2g_{i-1} - f g_{i-1}^2) \mod p^{2^i}$
3. return $g_r$

**Theorem 5.** *Algorithm 9.10 correctly computes the inverse of* $f \mod p^l$*. It uses* $O(M(l \log p))$ *word operations if* $R = \mathbb{Z}, p > 1$*, and* $|f| < p^l$*, and* $O(M(l \deg p))$ *operations in* $D$ *if* $R = D[x]$ *for a commutative ring* $D$*,* $p$ *is monic, and* $\deg f < l \deg p$*.*

## 9.2 Generalized Taylor expansion and radix conversion

### Algorithm 9.14 Generalized Taylor expansion

Input: $a, p \in R[x]$, $\deg p = m, \deg a < km$, $p$ is monic, $k$ is a power of 2.

Output: $a_0, \ldots, a_{k-1} \in R[x]$ such that $a = \sum_{0 \le i < k} a_i p^i, \deg a_i < m, 0 \le i < k$

1. if $k = 1$ then return $a_0 = a$
2. $t \longleftarrow k/2$ compute $p^t \in R[x]$, $q \longleftarrow a$ quo $p^t$, $r \longleftarrow a$ rem $p^t$.
3. compute recursively $r = \sum_{0 \le i < t} a_i p^i$ and $q = \sum_{0 \le i < t} a_{t+i} p^i$.
4. return $a_0, \ldots, a_{k-1}$.

**Corollary 2.** *Let $n \in \mathbb{N}$ be a power of 2. The Taylor expansion of a polynomial $a \in R[x]$ of degree $n$ around $u \in R$ can be computed using at most $(3M(n) + O(n)) \log n$ or $O(M(n) \log n)$ operations in $R$.*

**Theorem 6.** *Given $a, p \in \mathbb{N}$ with $p$ of length $m$ and $a$ of length at most $km$ for some $k, m \in \mathbb{N}$, we can compute the $p$-adic expansion of $a$ using $O(M(km) \log k)$ word operations.*

## 9.3 Formal derivatives and Taylor expansion

Let $R$ be an arbitrary ring(commutative, with 1). For $\varphi = \sum_{0\leq i\leq n} \varphi_i y^i \in R[y]$, we define the formal derivative of $\varphi$ by

$$\varphi' = \sum_{0\leq i\leq n} i\varphi_i y^{i-1}$$

**Lemma 1.** *1. $'$ is $R$-linear.* $(a\varphi + b\psi)' = a\varphi' + b\psi'$

*2. $'$ satisfies the Leibniz(product) rule* $(\varphi\psi)' = \varphi'\psi + \psi'\varphi$

*3. $'$ satisfies the chain rule* $(\varphi(\psi))' = \varphi'(\psi)\psi'$.

**Lemma 2.** *Let* $\varphi = \sum_{0 \le i \le n} \varphi_i y^i \in R[y]$. *Then*

$$\varphi = \varphi(g) + \varphi'(g)(y-g) + \psi \cdot (y-g)^2$$

*for some* $\psi \in R[y]$.

## 9.4 Solving polynomial equations via Newton iteration

**Lemma 3.** *(Quadratic convergence of Newton iteration) Let $\varphi \in R[y]$, $g, h \in R$ with $\varphi(g) \equiv 0 \mod p^k$ for some $k \in \mathbb{N}_{>0}$ and $\varphi'(g)$ invertible modulo $p$, and suppose that Newton's formula holds "approximately":*

$$h \equiv g - \varphi(g)\varphi'(g)^{-1} \mod p^{2k}$$

*Then $\varphi'(h)$ is invertible modulo $p$,*

$$\begin{aligned} \varphi(h) &\equiv 0 \mod p^{2k}, \\ h &\equiv g \mod p^k \end{aligned}$$

Proof:

1.

$$h \equiv g - \varphi(g)\varphi'(g)^{-1} \equiv g \mod p^k$$

2.

$$\begin{aligned}
\varphi(h) &= \varphi(g) + \varphi'(g)(h-g) + \psi(h-g)\cdot(h-g)^2 \\
&= \varphi(g) + \varphi'(g)(h-g) \\
&\equiv \varphi(g) + \varphi(g)' \cdot (-\varphi(g)\varphi'(g)^{-1}) \equiv 0 \mod p^{2k}
\end{aligned}$$

## Algorithm 9.22 $p$-adic Newton iteration

Input: $\varphi \in R[y]$, $p \in R, l \in \mathbb{N}_{>0}, g_0 \in R$ with $\varphi(g_0) \equiv 0 \mod p$ and $s_0 \equiv \varphi'(g_0) \mod p$.

Output: $g \in R$ with $\varphi(g) \equiv 0 \mod p^l$ and $g \equiv g_0 \mod p$.

1. $r \longleftarrow \lceil \log l \rceil$

2. for $i = 1, \ldots, r$ compute $g_i, s_i \in R$ such that

$$g_i \equiv g_{i-1} - \varphi(g_{i-1})s_{i-1} \mod p^{2^i},\ s_i \equiv 2s_{i-1} - \varphi'(g_i)s_{i-1}^2 \mod p^{2^i}$$

3. compute $g \in R$ with $g \equiv g_{r-1} - \varphi(g_{r-1})s_{r-1} \mod p^l$,
   return $g_r$

**Theorem 7.** *When* $R = D[x]$ *for a ring* $D$*(commutative, with 1),* $p = x, g_0 \in D, l \in \mathbb{N}$ *is a power of 2, and* $\varphi \in R[y]$ *with* $\deg_x \varphi < l$ *and* $\deg_x \varphi < 1$*, then the above algorithm takes* $(3n + 3/2)M(l) + O(nl)$ *operations in* $D$*.*

**Theorem 8.** *When* $R = \mathbb{Z}, 0 \leq g_0 < p$*, and* $\varphi$ *has degree* $n$ *and coefficients absolutely less than* $p^l$*, then algorithm 9.22 takes* $O(nM(l \log p))$ *word operations.*

**Theorem 9.** *Uniqueness of Newton iteration Let $\varphi \in R[y], g \in R$ with $\varphi(g) \equiv 0 \mod p$ and $\varphi'(g)$ invertible modulo $p$, and $l \in \mathbb{N}_{>0}$. If $h, h^* \in R$ are solutions modulo $p^l$ with*

$$\begin{aligned} h &\equiv g \equiv h^* \mod p \\ \varphi(h) &\equiv 0 \equiv \varphi(h^*) \mod p^l \end{aligned}$$

*then*

$$h \equiv h^* \mod p^l.$$

Proof:

$$\varphi(h^*) - \varphi(h) = (h^* - h)(\varphi'(h) + c \cdot (h^* - h)).$$

## 9.5 Computing integer roots

Assume $n$ is odd and $a$ is odd, $\varphi(y) = y^n - a$, $g_0 = 1$, $\varphi'(g_0) = n \cdot 1^{n-1} \equiv 1 \neq 0 \mod 2$. Choose $k \in \mathbb{N}$ minimal such that $2^{nk} > a$, $r = \lceil \log k \rceil$, run algorithm 9.22.

For saving time, set $t_0 = 1$ and compute $t_i = g_i^{n-1}$ rem $2^{2^{i+1}}$, compute

$$g_i \equiv g_{i-1} - \varphi(g_{i-1})s_{i-1} \equiv g_{i-1} - (g_{i-1}t_{i-1} - a)s_{i-1} \mod 2^{2^i}$$

$$s_i \equiv 2s_{i-1} - \varphi'(g_i)s_{i-1}^2 \equiv g_{i-1} - nt_is_{i-1}^2 \mod 2^{2^i}$$

# 9.6 Valuations, Newton iteration, and Julia sets