CS 487 / · · ·

Introduction to Symbolic Computation

University of Waterloo
Éric Schost
eschost@uwaterloo.ca
Newton iteration
Newton iteration is a way to compute approximate solutions to various problems. To compute a solution of $P(z) = 0$, we use the iteration

$$z_0 = \text{random}, \quad z_{i+1} = z_i - \frac{P(z_i)}{P'(z_i)}.$$
Example: computing $\sqrt{2}$

Take

$$P(z) = z^2 - 2,$$

so that $P'(z) = 2z$. Newton’s iteration is

$$z_{i+1} = z_i - \frac{z_i^2 - 2}{2z_i}.$$

Example with $z_0 = 2$

$$
\begin{align*}
  z_1 &= 1.51215959674934899234183093789 \ldots \\
  z_2 &= 1.41738565666481434293120019521 \ldots \\
  z_3 &= 1.41421711193003752379226117251 \ldots \\
  z_4 &= 1.41421356237754958211965910664 \ldots \\
  z_5 &= 1.41421356237309504880169573972 \ldots \\
  z_6 &= 1.41421356237309504880168872421 \ldots
\end{align*}
$$
There are strong analogies between real numbers

\[ a = 0.93493847630496 \ldots = \sum_{i \geq 1} a_i \frac{1}{10^i} \]

and power series

\[ S = \sum_{i \geq 0} s_i x^i. \]

- both are infinite expansions;
- computationally, we are interested in computing truncations at finite precision;
- similar techniques apply.

Remark: power series are easier to handle than real numbers, because there is no carry.
The example we saw can be vastly generalized and applied to power series computations.

**In a nutshell:**
- applies to compute exponential, inverses, logarithms, square roots, ... of power series,
- more generally, solutions of polynomial or differential equations.

**Main feature: efficiency!**
- typical behaviour: the number of correct terms doubles each step;
- combined with polynomial multiplication \(\Rightarrow\) quasi-optimal.
Let $M(d)$ denote the cost of polynomial multiplication in degree $d$:

- $M(d) \in O(d^2)$ for a naive algorithm
- $M(d) \in O(d^{1.6})$ for Karatsuba algorithm
- $M(d) \in O(d \log d)$ using Fast Fourier Transform (if the field has roots of 1)
- $M(d) \in O(d \log d \log \log d)$ using Fast Fourier Transform in general.

Technically, we ask $M(d + d') \geq M(d) + M(d')$. 
A few rules for estimating complexity

We know that

\[ 1 + 2 + 4 + \cdots + 2^{s-1} = 2^s - 1 \leq 2^s. \]

We have similar estimates for polynomial multiplication:

\[ M(1) + M(2) + M(4) + \cdots + M(2^{s-1}) \leq M(2^s). \]
A few rules for estimating complexity

We know that

\[ 1 + 2 + 4 + \cdots + 2^{s-1} = 2^s - 1 \leq 2^s. \]

We have similar estimates for polynomial multiplication:

\[ M(1) + M(2) + M(4) + \cdots + M(2^{s-1}) \leq M(2^s). \]

**Proof.**

\[
M(a) + M(b) \leq M(a + b) \implies 2M(a) \leq M(2a) \\
\implies 2^k M(a) \leq M(2^k a) \\
\implies 2^k M(2^s / 2^k) \leq M(2^s) \\
\implies M(2^s / 2^k) \leq 2^{-k} M(2^s)
\]
A few rules for estimating complexity

We know that

\[ 1 + 2 + 4 + \cdots + 2^{s-1} = 2^s - 1 \leq 2^s. \]

We have similar estimates for polynomial multiplication:

\[ \text{M}(1) + \text{M}(2) + \text{M}(4) + \cdots + \text{M}(2^{s-1}) \leq \text{M}(2^s). \]

**Proof.**

\[ \text{M}(a) + \text{M}(b) \leq \text{M}(a + b) \implies 2\text{M}(a) \leq \text{M}(2a) \]
\[ \implies 2^k\text{M}(a) \leq \text{M}(2^ka) \]
\[ \implies 2^k\text{M}(2^s/2^k) \leq \text{M}(2^s) \]
\[ \implies \text{M}(2^s/2^k) \leq 2^{-k}\text{M}(2^s) \]

**Corollary.** If \( T(2n) \leq T(n) + CM(n) \), then \( T(n) \in O(M(n)) \).
Inversion
Iteration for the inverse

Given a power series

\[ f = f_0 + f_1 x + f_2 x^2 + \cdots, \quad f_0 \neq 0 \]

we want to compute the coefficients of

\[ g = g_0 + g_1 x + g_2 x^2 + \cdots \]

such that \( fg = 1 \).

**Naive algorithm:**

- compute one term after the other, by identification.
- slow: \( O(n^2) \) operations for \( n \) terms.
- at least, this proves that \( g \) exists and is unique.
**Newton iteration**

**Idea:** $g$ satisfies $P(g) = 0$, with

$$P(z) = \frac{1}{z} - f.$$

Newton iteration is

$$z_0 = \frac{1}{f_0}, \quad z_{i+1} = 2z_i - f z_i^2.$$

**Claim:** $f z_i \mod x^{2i} = 1$
Newton iteration

Idea: $g$ satisfies $P(g) = 0$, with

$$P(z) = \frac{1}{z} - f.$$ 

Newton iteration is

$$z_0 = \frac{1}{f_0}, \quad z_{i+1} = 2z_i - fz_i^2.$$ 

Claim: $fz_i \mod x^{2^i} = 1$

Proof: induction on $i$ (OK for $i = 0$)

- Assume $fz_i \mod x^{2^i} = 1$. This means $fz_i = 1 + x^{2^i} R$.
- $fz_{i+1} = f(2z_i - fz_i^2) = 1 - (fz_i - 1)^2 = 1 - x^{2^{i+1}} R^2$. 
Example

Take

\[ f = 1 + 10x - 5x^2 + 6x^3 - 4x^4 + 10x^5 - 8x^6 + 9x^7 + \cdots \]

We get

\[ z_0 = 1 \]

\[ f z_0 = 1 + 10x + \cdots \]

\[ z_1 = 1 - 10x + 5x^2 - 6x^3 + 4x^4 - 10x^5 + 8x^6 - 9x^7 + \cdots \]

\[ f z_1 = 1 - 100x^2 + \cdots \]

\[ z_2 = 1 - 10x + 105x^2 - 1106x^3 + 1649x^4 - 2700x^5 + 3409x^6 - 6047x^7 + \cdots \]

\[ f z_2 = 1 - 10000x^4 + \cdots \]

\[ z_3 = 1 - 10x + 105x^2 - 1106x^3 + 11649x^4 - 122700x^5 + 1292409x^6 - 13613047x^7 + \cdots \]

\[ f z_3 = 1 + \cdots \]
Truncated computations

The \(z_i\)'s are too large. We need to compute with truncations.
Define the sequence

\[
\zeta_0 = \frac{1}{f_0}, \quad \zeta_{i+1} = (2\zeta_i - f \zeta_i^2) \mod x^{2i+1}.
\]

Claim: for all \(i\), \(\zeta_i = z_i \mod x^{2i}\)
Truncated computations

The $z_i$’s are too large. We need to compute with truncations. Define the sequence

$$
\zeta_0 = \frac{1}{f_0}, \quad \zeta_{i+1} = (2\zeta_i - f\zeta_i^2) \mod x^{2i+1}.
$$

**Claim:** for all $i$, $\zeta_i = z_i \mod x^{2i}$

**Proof:** induction on $i$ (OK for $i = 0$)
Take

\[ f = 1 + 10x - 5x^2 + 6x^3 - 4x^4 + 10x^5 - 8x^6 + 9x^7 + \cdots \]

We get

\[ \zeta_0 = 1 \]

\[ f \zeta_0 = 1 + 10x + \cdots \]

\[ \zeta_1 = 1 - 10x \]

\[ f \zeta_1 = 1 - 105x^2 + \cdots \]

\[ \zeta_2 = 1 - 10x + 105x^2 - 1106x^3 \]

\[ f \zeta_2 = 1 - 11649x^4 + \cdots \]

\[ \zeta_3 = 1 - 10x + 105x^2 - 1106x^3 + 11649x^4 - 122700x^5 + 1292409x^6 - 13613047x^7 \]

\[ f \zeta_3 = 1 + \cdots \]
Cost analysis

Cost of a single step:

- To get $\zeta_{i+1}$ from $\zeta_i$, we compute $2\zeta_i - f\zeta_i^2$ modulo $x^{2i+1}$.
- This costs $O(M(2^{i+1}))$.

Cumulated cost:

- To get $\zeta_i$ from $\zeta_0 = 1$, the cost is

$$O(M(2) + M(4) + \cdots + M(2^i)) = O(M(2^i)).$$

- In other words: to get $1/f \mod x^n$, the cost is $O(M(n))$. 
Algebraic power series
Roots of polynomial

Def.

• A power series

\[ f = \sum_{i \geq 0} f_i x^i \]

is algebraic if there exists a polynomial

\[ P(x, z) \] such that \( P(x, f) = 0. \)

Examples.

• rational power series \( f(x) = n(x)/d(x) \)

\[ P(x, z) = d(x)z - n(x) \]

• \( \sqrt{1 + x} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \frac{5}{128}x^4 + \frac{7}{256}x^5 + \cdots \)

\[ P(x, z) = z^2 - (1 + x) \]

• \( \exp(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots \) is not algebraic.
Let $f(x) = \sqrt{1 + x}$. The power series expansion of $f$ gives approximations of the curve $z^2 - (1 + x) = 0$ at $(0, 1)$. 
Let \( f(x) = \sqrt{1 + x} \). The power series expansion of \( f \) gives approximations of the curve \( z^2 - (1 + x) = 0 \) at (0, 1).

\[
f \mod x = 1
\]
Let $f(x) = \sqrt{1 + x}$. The power series expansion of $f$ gives approximations of the curve $z^2 - (1 + x) = 0$ at $(0, 1)$.

\[ f \mod x^2 = 1 + \frac{1}{2}x \]
Let $f(x) = \sqrt{1 + x}$. The power series expansion of $f$ gives approximations of the curve $z^2 - (1 + x) = 0$ at $(0, 1)$.

$$f \mod x^3 = 1 + \frac{1}{2}x - \frac{1}{8}x^2$$
A lot of sequences arising from enumeration problems satisfy nice properties, like having an algebraic generating series.

**Example:** Catalan numbers.

Let \( C_n \) be the number of binary trees with \( n \) nodes.

\[
C_0 = 1, \quad C_1 = 1, \quad C_2 = 2, \quad C_3 = 5, \ldots
\]
Recurrence relation

To build a tree with $n$ nodes, you

- set the root (so you have $n - 1$ nodes left)
- choose $p \leq n - 1$ and put $p$ nodes on the left
- and $n - 1 - p$ nodes on the right.

This gives

$$C_n = \sum_{p=0}^{n-1} C_p C_{n-1-p}$$

for $n \geq 1$. 
The generating series

Let \( f = \sum_{i \geq 0} C_i x^i \). Then,

\[
\sum_{p=0}^{n-1} C_p C_{n-1-p}
\]

is the coefficient of \( x^{n-1} \) in \( f^2 \).
Let \( f = \sum_{i \geq 0} C_i x^i \). Then,

\[
\sum_{p=0}^{n-1} C_p C_{n-1-p}
\]

is the coefficient of \( x^{n-1} \) in \( f^2 \).

Multiplying the recurrence relation by \( x^n \) and summing for \( n \geq 1 \), we get

\[
f - 1 = xf^2.
\]

So \( f \) is algebraic, with

\[
P(x, z) = xz^2 - z + 1
\]
In this case, we have an **explicit formula**

\[ f = \frac{1 - \sqrt{1 - 4x}}{2x}, \]

from which one can deduce

\[ C_n = \frac{1}{n + 1} \binom{2n}{n}. \]

In general, though, there is **no** closed formula.
Computing the expansion

We will compute the expansion of $f$ such that

$$P(x,f) = 0$$

subject to the following:

- the constant term $f_0$ of $f$ is known
  we need it to start the process
- the partial derivative $\frac{\partial P}{\partial z}(0,f_0)$ is not zero.
  at the starting point, the tangent to the curve exists, and is not vertical
The slow algorithm

Suppose that we know the first terms

\[ f_{\text{init}} = f_0 + \cdots + f_{n-1}x^{n-1}, \]

such that

\[ P(x, f_{\text{init}}) \mod x^n = 0. \]

**Basic step**

- we look for a *single* extra term \( f_n x^n \), to get

\[ f_{\text{next}} = f_0 + \cdots + f_{n-1}x^{n-1} + f_n x^n, \]

such that

\[ P(x, f_{\text{next}}) \mod x^{n+1} = 0. \]

- we get it by identification.
Getting the next term

**Claim.** The coefficient of $x^n$ in $P(x, f_{\text{next}})$ is

$$\text{known stuff} + \frac{\partial P}{\partial z}(0, f_0)f_n.$$ 

So we can **solve** for $f_n$.

**Proof.** Tedious calculation.

Computing the $n$th term requires at least $n$ operations, whence a **cumulated cost** of at least $n^2$ for $f_1, \ldots, f_n$ (disregarding the dependency in $d = \deg(P)$).
Newton iteration:

\[ \zeta_0 = f_0 \quad \zeta_{i+1} = \zeta_i - \frac{P(x, \zeta_i)}{\partial P/\partial \zeta(x, \zeta_i)} \mod x^{2^{i+1}}. \]

Prop.

- this correctly computes the expansion of \( f \);
- the cost is \( O(dM(n)) \) for order \( n \).