Automatic Adaptation of the Computing Precision

Nathalie Revol
INRIA, Université de Lyon
LIP, ENS Lyon, France
Nathalie.Revol@ens-lyon.fr

V Taylor Models Workshop, Toronto, 21-23 May 2008

Context

My field: interval arithmetic and arbitrary precision.

Recurring problems:

- how to detect that the current computing precision does not suffice?
- how to increase the current computing precision?

Today's problem: how to increase the current computing precision?

Ideally. . .

Let the exact solution of a given problem be x^*.

Goal: compute an approximation $\tilde{x}$ of the exact solution x^* with prescribed accuracy:

$$|\tilde{x} - x^*| \leq \epsilon.$$

The optimal computing precision to reach this accuracy is p^*.

In the real world. . .

Let the exact solution of a given problem be x^*.
Goal: compute an approximation $\tilde{x}$ of the exact solution x^* with prescribed accuracy:

$$|\tilde{x} - x^*| \leq \epsilon.$$

The optimal computing precision to reach this accuracy is p^* but p^* is unknown.

Method:

- try to compute with a precision p_0
- if the result is not accurate enough, change the precision to p_1
  try to compute with the precision p_1
- try with precisions $p_2 \ldots p_i \ldots p_n$
  until the precision $p_n \geq p^*$.

Assumptions

Let $\tilde{x}(p)$ be the approximation computed with precision p.
Let $\varepsilon(p)$ be the accuracy: $|\tilde{x}(p) - x^*| \leq \varepsilon(p)$.

$$\varepsilon(p) \rightarrow 0 \text{ as } p \rightarrow +\infty.$$

Notations

Optimal precision p^*	Computing time t^*
precision p_0	computing time t_0
precision p_1	computing time t_1
$\vdots$	$\vdots$
precision p_n	computing time t_n
finally $p_n \geq p^*$	total computing time $T = \sum_{i=0}^{n} t_i$
	overhead $= T/t^*$

Question

How to minimize the overhead?

In other words, if p_i is not large enough,
how should we choose p_{i+1}?

Motivation

Theoretical results exist...
but practical experiments + intuition contradict these results!

Motivation

Theoretical results exist. . .
but practical experiments + intuition contradict these results!

Approach:

- survey existing work
- analyse the theoretical result to determine where and when we disagree.

Outline

iRRAM

Müller 2000, based on Brattka and Hertling 1995

It is well-known that the asymptotic complexity of sequences of iterated computations is of the same order as the complexity of the last iteration.

Choice of the new precision:

$$p_i = p_0 + g\frac{f^i - 1}{f - 1}, \text{ with } g = 50 \text{ and } f = 1.25.$$

or equivalently

$$p_i - p_0 = g + f.(p_{i-1} - p_0).$$

the precision bound is incremented as above, the computation is restarted from the beginning.

Mathemagix

van der Hoeven 2006

Then we let $p_{i+1} > p_i$ be such that $t_{i+1} \simeq 2t_i$ and we replace our p_i-digit approximation by a p_{i+1}-digit approximation.
[. . .]
Consequently the total time is approximately twice the time of the last iteration and less than 4 times the optimal time.
More generally, the evaluation at different precisions $p_1, \ldots p_n$ requires at most four times the evaluation at the maximal precision $\max(p_1, \ldots, p_n)$.

MPFR

Fousse, Hanrot, Lefèvre, Pélissier, Théveny, Zimmermann et al.

In theory: increase the precision to be able to round correctly
take into account the probability of failure
increment the current precision p by $\log p$
average of the overhead OK: large overheads are very unlikely

In practice:
first iteration: just add one extra-word
next iterations: precision is a geometric series of ratio 1.5

Interval Newton using MPFI

Revol 2003

Context:
determine the zeros of a function with a prescribed accuracy using interval Newton and a given precision.

Adaptation of the computing precision:
if the accuracy is not reached,
double the precision
since the result will be twice more accurate.

No mention neither of the computing time nor of the overhead.

Reminder: Notations

Optimal precision p^* | Computing time t^*

---

precision p_0 | computing time t_0
precision p_1 | computing time t_1
$\vdots$ | $\vdots$
precision p_n | computing time t_n

---

finally $p_n \geq p^*$ total computing time $T = \sum_{i=0}^{n} t_i$

overhead $= T/t^*$

Theoretical results

Kreinovich and Rump, 2006

Result 1:
if precision p_i so that $t_i = 2t_{i-1}$
then overhead $= 4$.
Furthermore: 4 is the minimal overhead for every geometric sequence of (t_i).

Result 2:
for every increasing sequence of (t_i),
if the total time is $T = \sum_{i=0}^{n} t_i$
then overhead ≥ 4.

Questions

- Is this optimal result really optimal
- Can the overhead be < 4?
- Can the overhead be only 2 for Newton?
- Can the overhead be even less?

Outline

Assumptions

The new iteration can benefit from the previous computations, instead of being restarted from scratch.

Typical example: Newton

Starting from the last iterate and doubling the computing precision yields a result with twice the precision of the last iterate.

$$\begin{array}{lllllll} t^* = & t_0^* + t_1^* + \ldots + & t_i^* & + & t_{i+1}^* & + \ldots + & t_{n^*}^* & \\ T = & & t_0 & + & t_1 & + \ldots + & t_{n-1} & + t_n \end{array}$$

Overhead $\leq 1 + t_n / t^*$.

Example of Newton with time linear in the precision

Example: if $t_i = c.p_i^*$ and $p_i^* = 2p_{i-1}^*$,
optimal time $t^* = \sum_{i=0}^{n} c.p_i^* = \sum_{i=0}^{n} c.p^*/2^i = 2.c.p^*$:
time of the last iteration = time of all previous iterations.

If p^* is not known,
iterations with precisions $p_o, p_1, \dots p_{n-1} < p^*$, $p_n \geq p^*$.
At worse $p_n \simeq 2p^*$,
time for this extra-iteration = time of all previous iterations $\simeq$ optimal time.

In conclusion, total time $\simeq 2$ optimal time
i.e. **overhead** ≤ 2.

Notations

Ideal world	precision p_0^*	time t_0^*
	precision p_1^*	time t_1^*
	$\vdots$	$\vdots$
	precision p_m^*	time t_m^*
	$p^* := p_m^*$	total computing time $T^* = \sum_{i=0}^m t_i^*$
Real world	precision p_0	computing time t_0
	precision p_1	computing time t_1
	$\vdots$	$\vdots$
	precision p_n	computing time t_n
	finally $p_n \geq p^*$	total computing time $T = \sum_{i=0}^n t_i$
		overhead $= T/T^*$

Simplifying Assumptions

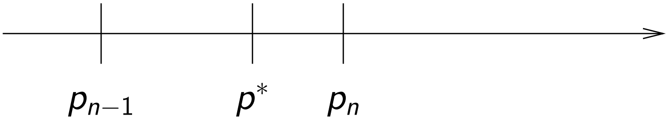
Computing times are proportional to the computing precision: $t = p$.
Precisions are equal except the last one:

$$p_0 = p_0^*,\ p_1 = p_1^*, \ldots p_{n-1} < p_m^* \leq p_n.$$

Thus

$$T - T^* = t_n - t^* \leq t_n - t_{n-1}.$$

Important point



Asymptotically optimal results

Idea: inspired from amortized techniques by Floyd and Brent
adapted from [Beaumont, Daoudi, Maillard, Manneback, Roch].

Principle: increase p_i at a slower pace as i increases
so as no to overtake p^* by too far:

$$p_i = \rho^{f(i)}.p_0$$

where $\rho > 1$ and $f(i) \to \infty$ when $i \to \infty$,
but not too quickly: $f(i)/i \to 0$ when $i \to \infty$.

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$T - T^* \leq t_n - t_{n-1}$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$\begin{aligned} T - T^* &\leq t_n - t_{n-1} \\ &\leq \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \end{aligned}$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$
\begin{aligned}
T - T^* &\leq t_n - t_{n-1} \\
&\leq \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\
&\leq \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1)
\end{aligned}
$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$
\begin{aligned}
T - T^* &\leq t_n - t_{n-1} \\
&\leq \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\
&\leq \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1) \\
&\leq \rho^{(n-1)^\alpha} \cdot \rho^{\alpha \cdot n^{\alpha-1} + o(n^{\alpha-1})}
\end{aligned}
$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$\begin{aligned} T - T^* &\leq t_n - t_{n-1} \\ &\leq \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\ &\leq \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1) \\ &\leq \rho^{(n-1)^\alpha} \cdot \rho^{\alpha \cdot n^{\alpha-1} + o(n^{\alpha-1})} \end{aligned}$$

When $n \to \infty$, since $0 < \alpha < 1$, $\alpha \cdot n^{\alpha-1} \to 0$ and $\rho^{\alpha \cdot n^{\alpha-1}} \to 1$.

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$\begin{array}{rcl} T - T^* & \leq & t_n - t_{n-1} \\ & \leq & \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\ & \leq & \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1) \\ & \leq & \rho^{(n-1)^\alpha} \cdot \rho^{\alpha \cdot n^{\alpha-1} + o(n^{\alpha-1})} \end{array}$$

When $n \to \infty$, since $0 < \alpha < 1$, $\alpha \cdot n^{\alpha-1} \to 0$ and $\rho^{\alpha \cdot n^{\alpha-1}} \to 1$.
Thus

$$T - T^* \sim \rho^{(n-1)^\alpha} < p^*$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$\begin{array}{rcl} T - T^* & \leq & t_n - t_{n-1} \\ & \leq & \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\ & \leq & \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1) \\ & \leq & \rho^{(n-1)^\alpha} \cdot \rho^{\alpha \cdot n^{\alpha-1} + o(n^{\alpha-1})} \end{array}$$

When $n \to \infty$, since $0 < \alpha < 1$, $\alpha \cdot n^{\alpha-1} \to 0$ and $\rho^{\alpha \cdot n^{\alpha-1}} \to 1$.
Thus

$$T - T^* \sim \rho^{(n-1)^\alpha} < p^*$$

$$\frac{T}{T^*} - 1 < 1$$

First example: $f(i) = i^\alpha$ with $0 < \alpha < 1$

$$\begin{array}{rcl} T - T^* & \leq & t_n - t_{n-1} \\ & \leq & \rho^{n^\alpha} - \rho^{(n-1)^\alpha} \\ & \leq & \rho^{(n-1)^\alpha} \cdot (\rho^{n^\alpha - (n-1)^\alpha} - 1) \\ & \leq & \rho^{(n-1)^\alpha} \cdot \rho^{\alpha \cdot n^{\alpha-1} + o(n^{\alpha-1})} \end{array}$$

When $n \to \infty$, since $0 < \alpha < 1$, $\alpha \cdot n^{\alpha-1} \to 0$ and $\rho^{\alpha \cdot n^{\alpha-1}} \to 1$.
Thus

$$T - T^* \sim \rho^{(n-1)^\alpha} < p^*$$

$$\frac{T}{T^*} - 1 < 1$$

and finally

$$\frac{T}{T^*} < 2.$$

Second example: $f(i) = i/\log i$

$$T - T^* \leq t_n - t_{n-1}$$

Second example: $f(i) = i / \log i$

$$\begin{aligned} T - T^* &\leq t_n - t_{n-1} \\ &\leq \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \end{aligned}$$

Second example: $f(i) = i/\log i$

$$
\begin{array}{rcl}
T - T^* & \leq & t_n - t_{n-1} \\
 & \leq & \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \\
 & \leq & \rho^{\frac{n-1}{\log(n-1)}} \cdot (\rho^{\frac{n}{\log n} - \frac{n-1}{\log(n-1)}} - 1)
\end{array}
$$

Second example: $f(i) = i/\log i$

$$\begin{aligned} T - T^* &\leq t_n - t_{n-1} \\ &\leq \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \\ &\leq \rho^{\frac{n-1}{\log(n-1)}} \cdot (\rho^{\frac{n}{\log n} - \frac{n-1}{\log(n-1)}} - 1) \end{aligned}$$

Since $\frac{n}{\log n} - \frac{n-1}{\log(n-1)} < \frac{1}{\log(n-1)}$

$$T - T^* \leq \rho^{(n-1)^\alpha} \cdot (\rho^{\frac{1}{\log(n-1)}} - 1)$$

Second example: $f(i) = i/\log i$

$$
\begin{aligned}
T - T^* &\leq t_n - t_{n-1} \\
&\leq \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \\
&\leq \rho^{\frac{n-1}{\log(n-1)}} \cdot \left(\rho^{\frac{n}{\log n} - \frac{n-1}{\log(n-1)}} - 1\right)
\end{aligned}
$$

Since $\frac{n}{\log n} - \frac{n-1}{\log(n-1)} < \frac{1}{\log(n-1)}$

$$
\begin{aligned}
T - T^* &\leq \rho^{(n-1)^\alpha} \cdot \left(\rho^{\frac{1}{\log(n-1)}} - 1\right) \\
&\leq \rho^{(n-1)^\alpha} \cdot \frac{1}{\log(n-1)}
\end{aligned}
$$

using this lemma: $n - 1 \sim \log_\rho p^* \cdot \log\log_\rho p^*$ as $p^* \to \infty$.

Second example: $f(i) = i/\log i$

$$\begin{aligned} T - T^* &\leq t_n - t_{n-1} \\ &\leq \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \\ &\leq \rho^{\frac{n-1}{\log(n-1)}} \cdot (\rho^{\frac{n}{\log n} - \frac{n-1}{\log(n-1)}} - 1) \end{aligned}$$

Since $\frac{n}{\log n} - \frac{n-1}{\log(n-1)} < \frac{1}{\log(n-1)}$

$$\begin{aligned} T - T^* &\leq \rho^{(n-1)^\alpha} \cdot (\rho^{\frac{1}{\log(n-1)}} - 1) \\ &\leq \rho^{(n-1)^\alpha} \cdot \frac{1}{\log(n-1)} \end{aligned}$$

using this lemma: $n - 1 \sim \log_\rho p^* \cdot \log \log_\rho p^*$ as $p^* \to \infty$.
Thus

$$\frac{T}{T^*} - 1 < \frac{1}{\log(n-1)}$$

Second example: $f(i) = i / \log i$

$$
\begin{aligned}
T - T^* &\leq t_n - t_{n-1} \\
&\leq \rho^{\frac{n}{\log n}} - \rho^{\frac{n-1}{\log(n-1)}} \\
&\leq \rho^{\frac{n-1}{\log(n-1)}} \cdot (\rho^{\frac{n}{\log n} - \frac{n-1}{\log(n-1)}} - 1)
\end{aligned}
$$

Since $\frac{n}{\log n} - \frac{n-1}{\log(n-1)} < \frac{1}{\log(n-1)}$

$$
\begin{aligned}
T - T^* &\leq \rho^{(n-1)^\alpha} \cdot (\rho^{\frac{1}{\log(n-1)}} - 1) \\
&\leq \rho^{(n-1)^\alpha} \cdot \frac{1}{\log(n-1)}
\end{aligned}
$$

using this lemma: $n - 1 \sim \log_\rho p^* \cdot \log \log_\rho p^*$ as $p^* \to \infty$.
Thus

$$
\begin{aligned}
\frac{T}{T^*} - 1 &< \frac{1}{\log(n-1)} \\
\frac{T}{T^*} &\to 1 \text{ as } n \to \infty.
\end{aligned}
$$

General case

Remaining work: take into account the different paces in the progressions of p_i^* and p_j.

General case

Remaining work: take into account the different paces in the progressions of p_i^* and p_j.

Idea: one iteration using the optimal p_i^* requires several iterations using the p_j, to reach a sufficient precision.

General case

Remaining work: take into account the different paces in the progressions of p_i^* and p_j.

Idea: one iteration using the optimal p_i^* requires several iterations using the p_j, to reach a sufficient precision.

Recommendation: use large ρ.

Discussion

Large ρ...
If p^* small and ρ large: p_n much larger than p^*.

Assumptions: ρ must be large, p^* must be large, p_n must be large and p_n should not be much larger than p^*.

Asymptotical result...

Conclusion and future work

When adaptive precision is used,

- optimal overhead $= 4$ when computations must be restarted
- optimal overhead smaller otherwise:
  smaller than 2 with doubling precision
  asymptotically made as small as desired.

To do:

- Detect automatically when the precision does not suffice: compute with current precision and doubled precision to guess the need of higher precision.
- Imagine incremental computations.