Range reduction

Nicola Seriani

The Abdus Salam International Centre for Theoretical Physics,
Strada Costiera 11, 34151 Trieste, Italy

Introduction

- Up to now we have seen several approximation and interpolation methods
- They usually describe the function well in an interval of values of X
- It is difficult that these methods work properly for any choice of the X for which the function has to be evaluated
- It is then necessary to operate a **range reduction**, i.e. we limit our description of the function a-priori to a finite interval

See also: Jean-Michel Muller, Elementary functions: algorithms and implementations

Range reduction

- To evaluate a function f(x) for any x, we have to find a transformation
- $x \rightarrow x^*$
- $f(x) \rightarrow g(x^*)$
- such that we can deduce the value f(x) from $g(x^*)$

See also: Jean-Michel Muller, Elementary functions: algorithms and implementations

Range reduction

x

↓

x^*

(reduced argument), such that x^* belongs to a domain where g can be easily evaluated

↓

$g(x^*)$

↓

$f(x)$

Range reduction

- There are two kinds of reductions:
- Additive reduction: $x^* = x - k\,C$, where k is an integer and C is a constant (e.g., for trigonometric functions C is a multiple of $\pi/4$)
- Multiplicative reduction: $x^* = x/C^k$, where k is an integer and C is a constant (useful e.g. for logarithms)

Range reduction: example 1

- We need to evaluate $f(x) = \cos(x)$, but our algorithm to calculate it is valid only in the interval $[-\pi/2;+\pi/2]$. We choose $C = \pi$, and the computation of $\cos(x)$ is decomposed in three steps:

1) Compute x^* and k such that x^* belongs to $[-\pi/2;+\pi/2]$ and $x^* = x - k\pi$ (**additive reduction**)

2) Compute $g(x^*) = \cos(x^*)$

3) Reconstruct $f(x)$

Range reduction: example 1

1) Compute x^* and k such that x^* belongs to $[-\pi/2;+\pi/2]$ and $x^* = x - k\pi$ (**additive reduction**)

2) Compute $g(x^*) = \cos(x^*)$

3) Reconstruct $f(x)$: $\cos(x) = \cos(x^*+k\pi) = \cos(x^*)\cos(k\pi)-\sin(x^*)\sin(k\pi)$

Remember: $\cos(k\pi)=(-1)^k$; $\sin(k\pi)=0$

This gives

$$\cos(x)=(-1)^k \cos(x^*)$$

Range reduction: example 1

Computational cost

Without range reduction:

1) cos(x)
   (built-in function)

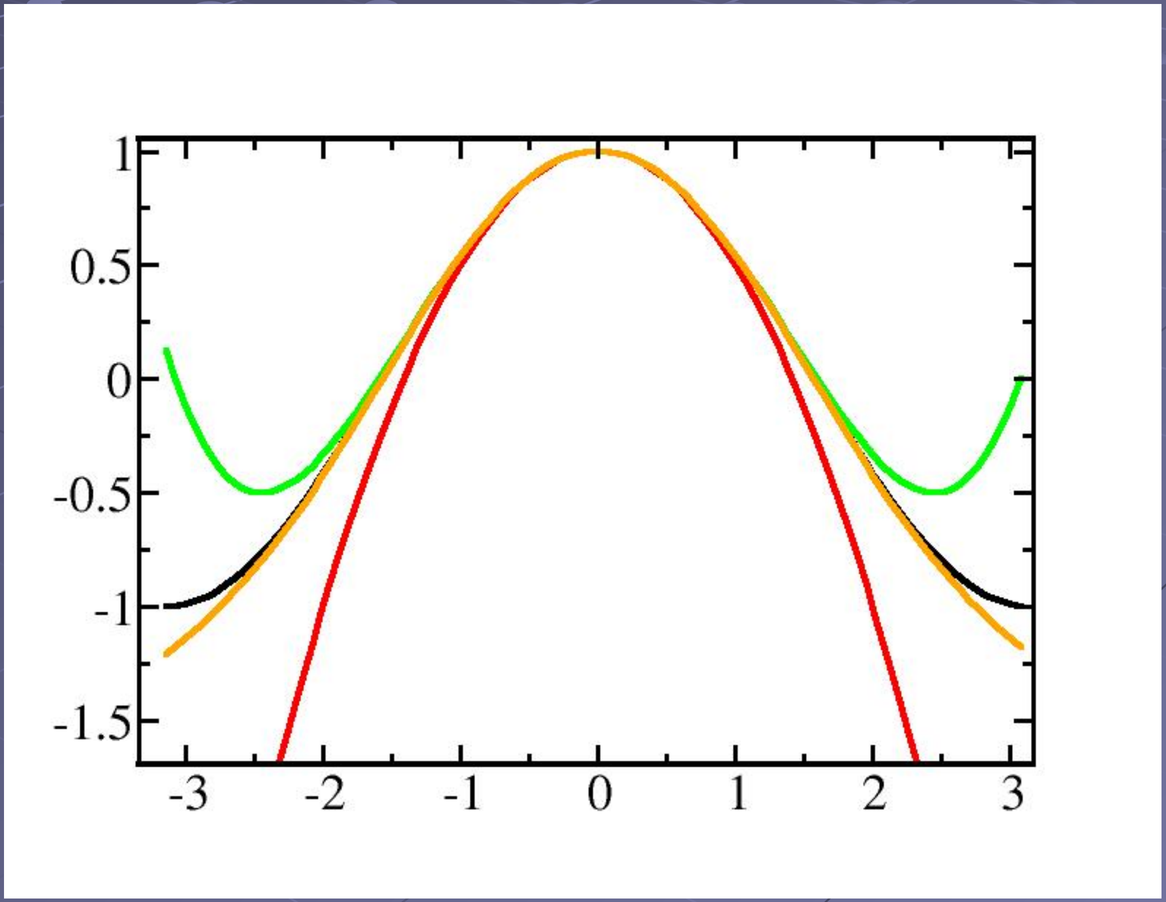
With range reduction:

1) Compute x* and k such that x* belongs to [-π/2;+π/2] and x* = x – kπ (Division by π, evaluation of integer part, subtraction)
2) Compute g(x*) = cos(x*) (e.g., Taylor)
3) Reconstruct f(x): $\cos(x)=(-1)^k \cos(x^*)$ (Multiplication by $(-1)^k$)

Range reduction: example 1

Computational cost

Computational cost of the Taylor explansion depends on range



Range reduction: example 1

Computational cost of the Taylor explansion depends on range: better to reduce to [−π/4;π/4]

Example 1 Computation of the cosine function. Assume that we want to evaluate $\cos(x)$, and that the convergence domain of the algorithm used to evaluate the sine and cosine of the reduced argument contains $[-\pi/4, +\pi/4]$. We choose $C = \pi/2$, and the computation of $\cos(x)$ is decomposed in three steps:

- Compute x^* and k such that $x^* \in [-\pi/4, +\pi/4]$ and $x^* = x - k\pi/2$
- Compute $g(x^*, k) =$

$$\begin{cases} \cos(x^*) & \text{if } k \bmod 4 = 0 \\ -\sin(x^*) & \text{if } k \bmod 4 = 1 \\ -\cos(x^*) & \text{if } k \bmod 4 = 2 \\ \sin(x^*) & \text{if } k \bmod 4 = 3 \end{cases} \tag{1}$$

- Obtain $\cos(x) = g(x^*, k)$

Daumas et al., Journal of Universal Computer Science 1, 162 (1995)

Range reduction: example 2

- We need to evaluate f(x) = ln(x), but our algorithm to calculate it is valid only in the interval [1/2;1].

1) Compute x^* and k such that x^* belongs to [1/2;1] and $x^* = x/2^k$ (**multiplicative reduction**)

2) Compute $g(x^*) = \ln(x^*)$

3) Compute $\ln(x) = g(x^*) + k \ln(2)$

Range reduction: example 3

- We need to evaluate $f(x) = e^x$

1) Compute $x^* = x - [x]$; $k = [x]$; k integer

2) Compute $g(x^*) = \exp(x^*)$

3) Compute $\exp(x) = \exp(x^*)\exp(k)$; $\exp(k)$ from multiplications, or from look-up tables

Daumas et al., Journal of Universal Computer Science 1, 162 (1995)