Round-off error analysis for a simple cumulative average

No Author Given

No Institute Given

1 Introduction

This short paper details a naive round-off error analysis for a simple cumulative average (SCA).

2 Problem Formulation

An informal description of the simple cumulative average is given as follows.

In a cumualtive average, we are concerned with computing the average of a sequence of n values $x_1, \cdot, x_n$ supposing that we already know the average of the sequence $x_1, \cdot, x_{n-1}$ and that the value x_n has only just become known to us.

If CA_{n-1} is the average of the sequence of values $x_1, \cdot, x_{n-1}$, then we can compute the average A_n as

$$CA_n = CA_{n-1} + \frac{x_n - CA_{n-1}}{n} \tag{1}$$

We observe that there are strictly more floating-point operations associated with computing a cumulative average than with computing an average. We are interested in deriving a practical upper bound on the forward absolute round-off error associated with such a computation, and showing that this bound holds for a specific C implementation of the SCA.

3 Rounding Error Analysis

3.1 Background

The standard model for floating-point error analysis represents the error in each operation between two exactly representable finite floating-point numbers x and y as

$$(x \ \mathtt{op}_{\mathbb{F}} \ y) = (x \ \mathtt{op}_{\mathbb{R}} \ y)(1 + \delta) + \epsilon \tag{2}$$

where ϵ and δ are small constants dendent on the precision of the computation.

3.2 Model problem

We consider a simple model problem where we assume the following knowns: (1) the total number of values being averaged and (2) an absolute upper bound on the values being averaged. While assumption (1) should be removed in future analysis, we argue that (2) is a reasonable assumption for most scientific and engineering domains where the values being averaged should represent some physical measurements. We also assume that all values being summed are exaclty representable in working-precision of the implementation.

If we define $CA_{\mathbb{F}}$ as the cumulative average carried out in floating-point arithmetic and $CA_{\mathbb{R}}$ as the cumulative average carried out in real arithmetic, then we would like to prove a bound of the form

$$|CA_{\mathbb{R}} - CA_{\mathbb{F}}| \leq \alpha \; n \; |CA_{\mathbb{R}}| \tag{3}$$

where n is the number of values being averaged and *alpha* is of order machine epsilon (`eps`).

For our model problem, we assume that we are computing the average of 10 elements, each element being positive and upper bounded by 10.

Functional models In order to prove that equation (3) holds for a specific C implementation of the cumulative average, we define *functional models* in Coq that correspond to functional programs for computing the cumulative average in real arithmetic and floating-point arithmetic, and then prove that the C program correctly implements the floating-point functional model.

```
Inductive mean_rel_R (g : R) : list R → R → Prop :=
| mean_rel_R_nil : mean_rel_R g [ ] g
| mean_rel_R_cons : forall mu m ms,
    mean_rel_R g ms mu →
    mean_rel_R g (m::ms) (mean_R_step mu m (INR (length ms))).

Inductive mean_rel_F (g : ftype Tsingle ) : list (ftype Tsingle ) → ftype Tsingle → Prop :=
| mean_rel_F_nil : mean_rel_F g [ ] g
| mean_rel_F_cons : forall mu m ms,
    mean_rel_F g ms mu →
    mean_rel_F g (m::ms) (mean_F_step mu m (Zconst Tsingle (Z.of_nat (length ms)))).

Definition mean_R_step (mu m n: R) : R := mu + (m − mu) / (1 + n).

Definition mean_F_step (mu m n: ftype Tsingle) : ftype Tsingle :=
  (mu + (m − mu) / (1 + n))%F32.
```