

УДК (519.6)

# АВТОМАТИЧЕСКАЯ ОЦЕНКА ПОГРЕШНОСТИ ВЫЧИСЛЕНИЙ В ЗАВИСИМОСТИ ОТ ПОГРЕШНОСТИ НАЧАЛЬНЫХ ДАННЫХ И ОКРУГЛЕНИЯ В РАСПАРАЛЛЕЛИВАЮЩЕЙ СИСТЕМЕ

Д.Н. Черданцев

*Государственное образовательное учреждение высшего профессионального образования «Ростовский государственный университет»*

Россия, 344006, Ростов-на-Дону, Б. Садовая ул., 105

E-mail: [denis\\_tch@mail333.com](mailto:denis_tch@mail333.com)

**Ключевые слова:** погрешность вычислений, погрешность начальных данных, погрешность округления, ОРС

**Key words:** calculation error, initial data error, rounding error, OPS

В Открытой распараллеливающей системе реализован режим компиляции с автоматической оценкой погрешности вычислений в зависимости от погрешности начальных данных и округления. Также рассматриваются преобразования программ: протягивание интервальных констант, частичные (смешанные) интервальные вычисления.

**AUTOMATIC ESTIMATION OF CALCULATION ERROR SUBJECT TO INITIAL DATA AND ROUNDING ERRORS IN PARALLELIZING SYSTEM**  
/ D.N. Cherdantsev (Rostov State University, 105 B. Sadovya, Rostov-na-Donu 344006, Russia, E-mail: [denis\\_tch@mail333.com](mailto:denis_tch@mail333.com)). Compilation mode with automatic estimation of calculation error subject to initial data and rounding errors was engineered in Open parallelizing system. Program transformation such as interval constant propagation and partial interval calculation are also considered.

Параллельные методы вычислений применяются в задачах с большим объемом вычислений. В таких задачах очень актуальна проблема оценки погрешности вычислений. Следует отметить, что ручной процесс вычисления оценок погрешности иногда очень трудоемок.

В рамках Открытой распараллеливающей системы (ОРС) [1] реализован режим компиляции с автоматической оценкой погрешности вычислений в зависимости от погрешности начальных данных и округления. Получен конвертор, преобразующий пользовательскую программу на языке Си в новую программу, которая помимо запрограммированных пользователем вычислений выдает и оценки погрешностей проведенных вычислений.

Для расчета погрешности, вносимой погрешностью начальных данных, производится замена входной переменной парой чисел (например, переменная  $A$  заменяется парой  $(A, A_{err})$ , где  $A$  – приближение к истинному значению, а  $A_{err}$  – оценка модуля максимальной относительной погрешности) и переопределение арифметических операций.

Операции переопределяются следующим образом [4]:

$$\begin{aligned}
 A + B &= (A + B, \frac{A}{A + B} A_{-err} + \frac{B}{A + B} B_{-err}), \\
 A - B &= (A + B, \frac{A}{A - B} A_{-err} + \frac{B}{A - B} B_{-err}), \\
 A * B &= (A * B, A_{-err} + B_{-err}), \\
 A / B &= (A / B, A_{-err} + B_{-err}).
 \end{aligned}$$

Для языков, не поддерживающих переопределение операций, в программу необходимо добавить операторы, выполняющие расчет относительной погрешности.

Для учета ошибок округления рассматривается симметричное округление, абсолютное значение относительной погрешности которого для десятичных чисел имеет вид:

$$r = 5 * 10^{-t},$$

где  $t$  – количество значащих цифр.

Для ЭВМ, работающими с двоичными числами, аналогично имеем:

$$r = 2^{-t}.$$

В итоге для учета обоих видов погрешностей операции переопределяются следующим образом:

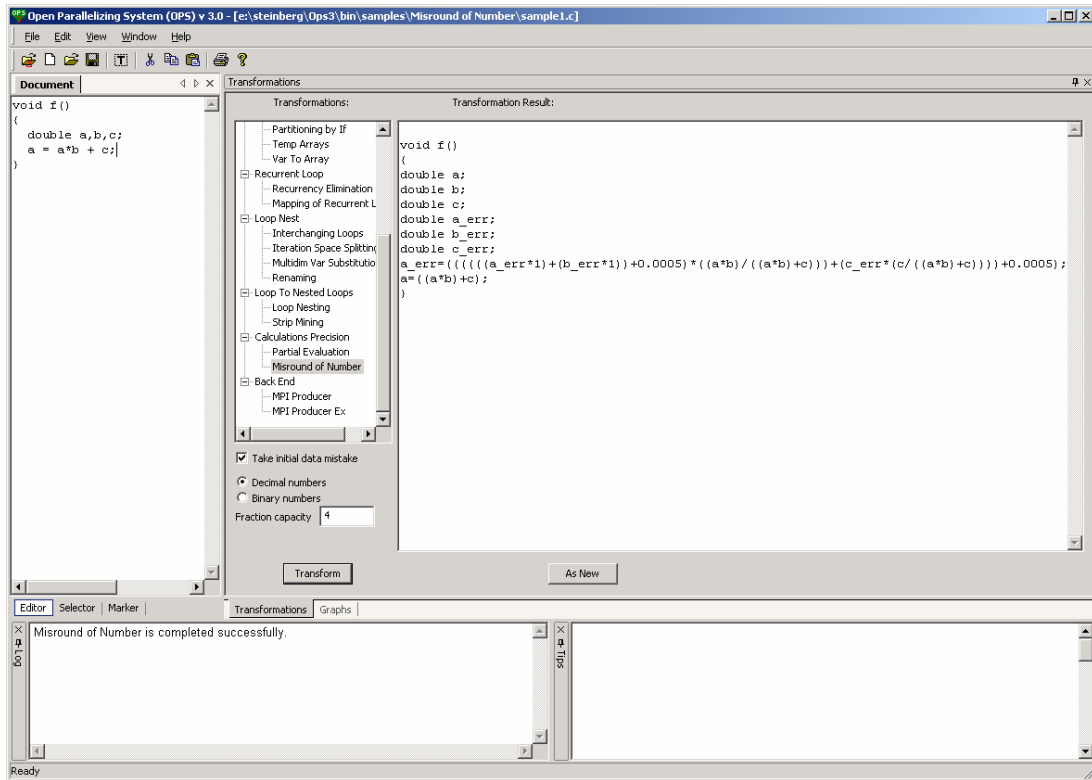
$$\begin{aligned}
 A + B &= (A + B, \frac{A}{A + B} A_{-err} + \frac{B}{A + B} B_{-err} + r), \\
 A - B &= (A + B, \frac{A}{A - B} A_{-err} + \frac{B}{A - B} B_{-err} + r), \\
 A * B &= (A * B, A_{-err} + B_{-err} + r), \\
 A / B &= (A / B, A_{-err} + B_{-err} + r).
 \end{aligned}$$

Основным минусом такого подхода является увеличение количества вычислений.

Пример работы ОРС приведен на рис. 1. В данном случае для оператора

$$a = a * b + c$$

(левое окно программы) был построен дополнительный оператор, вычисляющий погрешность (правое окно программы) с учетом наличия погрешности начальных данных и погрешности округления при 4-х значащих десятичных цифрах.



**Рис. 1.** Экран программы ОРС. Для оператора  $a = a*b + c$  (левое окно программы) был построен дополнительный оператор, вычисляющий погрешность (правое окно программы) с учетом наличия погрешности начальных данных и погрешности округления при 4-х значащих десятичных цифрах.

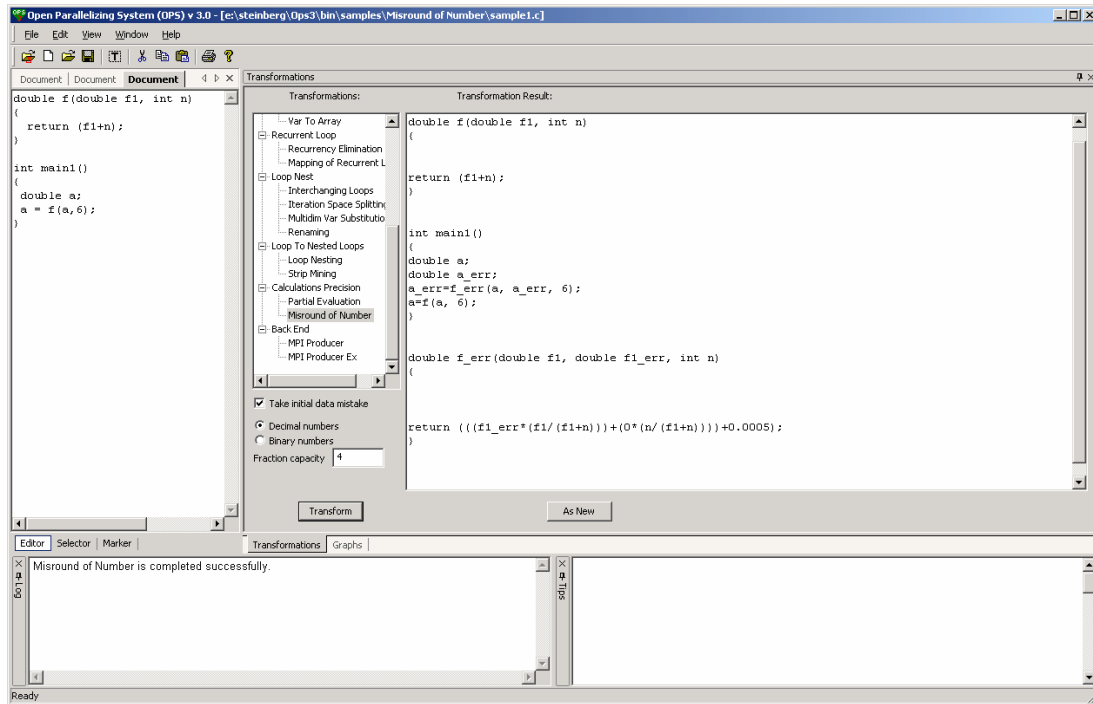
Следует отметить, что в ОРС есть возможность для получения оценки значения абсолютной ошибки, которая для текущего примера имеет вид:

$$\begin{aligned} &(((a\_err * 1 + b\_err * 1 + 0.0005) * (a * b)) / (a * b + c) + \\ &+ c\_err * (c / (a * b + c)) + 0.0005) * \\ & * (a * b + c). \end{aligned}$$

ОРС средствами упрощения выражений и приведения подобных преобразует это выражение к виду:

$$(a\_err + b\_err + 0.001) * (a * b) + (c\_err + 0.0005) * c.$$

При построении формул оценок для выражений, содержащих вызовы функций, иногда требуется больше усилий. Если в исходном выражении присутствует вызов функция  $f$ , то для оператора, оценивающего погрешность, будет построена дополнительная функция  $f\_err$ . Функция  $f\_err$  – оценка ошибки функции  $f$ . При этом если у функции  $f$  были формальные параметры, которые могут иметь погрешность, то у функции  $f\_err$  кроме этих параметров будут еще параметры, для передачи значений погрешности. В теле функции  $f\_err$  будут переопределены операции. Пример работы ОРС для вызова функции показан на рис. 2.



**Рис. 2.** Пример построения оценок погрешности вычислений для случая наличия в исходном выражении вызова функции. В окне слева находится текст исходной программы. В окне справа – преобразованный текст программы. Добавлен оператор для расчета погрешности переменной *a* и новая функция *f\_err* – для расчета погрешности функции *f*.

Так же отметим, что при таком преобразовании программы меняется ввод данных. Например, если в программе была одна переменная *A*, и вводилось ее значение, то теперь должно вводиться еще и значение для начальной погрешности. Проблема ввода существенно усложняется, если строятся погрешности для массивов, имеющих для каждого элемента различную начальную погрешность.

В ОРС реализовано преобразование программы протягивание интервальных констант. Это преобразование подставляет значение и оценку погрешности вместо имени константы в исходный оператор и оператор расчета погрешности. В результате такого преобразования исчезают некоторые дуги информационной зависимости, что может способствовать распараллеливанию.

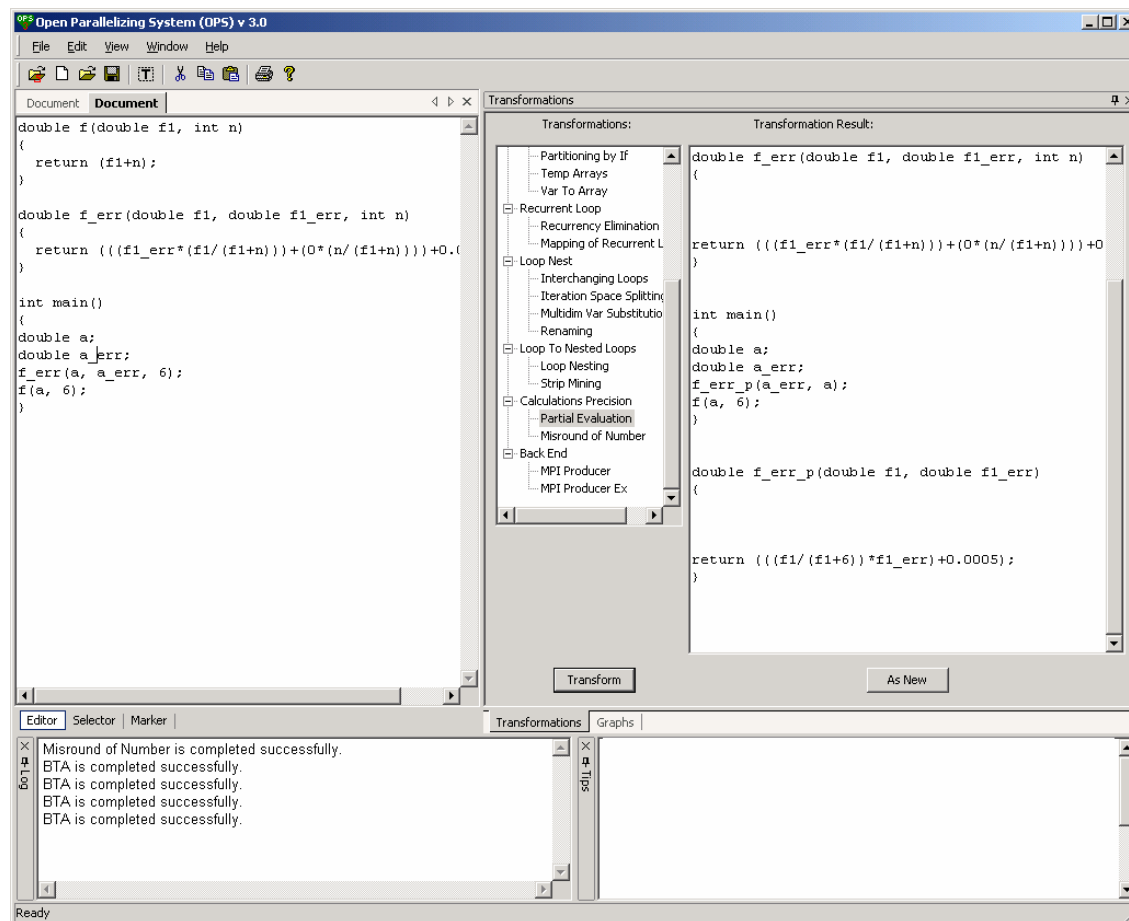
Пример. Данный фрагмент программы

```
c13_err = 0.001;
c13 = 10;
...
y_err = z_err[1+k] + c13_err;
y = z[1+k] + c13;
```

равносилен следующему

```
c13_err=0;
c13=10;
...
y_err=z_err[ (1+k) ]+0.001;
y=z[ (1+k) ]+10;
```

В ОРС реализовано преобразование программы моновариантные частичные интервальные вычисления [5, 6]. При этом учитывается протягивание интервальных констант. А при попадании интервальных констант (например, после протягивания) в параметры вызовов процедур или функций происходит формирование новых частично вычисленных процедур или функций.



**Рис. 3.** Пример применения преобразования частичные вычисления. В основной программе (функция main) вызов функции f\_err(a, a\_err, 6) заменен вызовом частично вычисленной функции f\_err\_p(a, a\_err). Частично вычисленная функция была сгенерирована автоматически, ее код представлен в нижней части правого окна программы.

## Список литературы

1. Штейнберг Б.Я., Макошенко Д.В., Черданцев Д.Н., Шульженко А.М. Внутреннее представление в Открытой распараллеливающей системе // Искусственный интеллект. Научно-теоретический журнал. Институт проблем искусственного интеллекта НАНУ. Украина, Донецк, ДонДИШИ, «Наука и Освита». 2003. № 4. С. 89-97.
2. Алефельд Г., Херцбергер Ю. Введение в интервальные вычисления / Пер. с англ. М.: Мир, 1987. 360 с.
3. Матиясевич Ю.В. Вещественные числа и ЭВМ // Сборник «Кибернетика и вычислительная техника» / Под ред. В.А. Мельникова. Вып. 2. М.: Наука, 1986. С. 104-133.
4. Мак-Кракен Д., Дорн У. Численные методы и программирование на Фортране / Пер. с англ. М.: Мир, 1977. 584 с.
5. Jones N.D., Gomard C.K., Sestoft P. Partial Evaluation and Automatic Program Generation. Prentice Hall, 1993.

6. Ершов А.П. Смешанные вычисления: потенциальные применения и проблемы исследования. <http://ershov.iis.nsk.su/archive/eaindex.asp?lang=1&did=2596>.
7. Векторизация программ // Векторизация программ: теория, методы, реализация / Сборник переводов статей. М.: Мир, 1991. С. 246-267.