ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ДЛЯ ТЕСТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОГРАММНЫХ МОДУЛЕЙ

А.Н. Коварцев, Е.Е. Серповская

Самарский государственный аэрокосмический университет им. С.П. Королёва

Рассматривается метод глобального поиска разрывов второго рода функции для поиска ошибок, имеющих «арифметический» характер. Представлена математическая модель поиска бесконечных разрывов функции. Приводится оценка качества предлагаемого алгоритма.

Введение

Основной целью тестирования программных модулей (ПМ) является обнаружение и устранение программных ошибок или (и), если это возможно, доказательство отсутствия последних в программе.

Техника тестирования программных модулей во многом определяется их специализацией. Рассмотрим ПМ, построенные на основе численного анализа и вычислительной математики. При построении таких программ в качестве «неделимых» программных единиц обычно выступают модули, реализующие функциональные преобразования между *n* и *m*-мерными пространствами действительных (комплексных) чисел. К предметным областям с такой схемой построения исходных модулей относятся САПР технических объектов, АСНИ и т.д.

Формально вычислительный модуль можно рассматривать как отображение

$$f_k : X_k^{in} \to X_k^{out}, \qquad (1)$$

действующее из множества входных параметров $X_k^{in} \in \Omega_{f_k}$ модуля на множество вычисляемых данных X_k^{out}.

Ошибочные ситуации в программных модулях возникают по разным причинам [1, 2]. Наиболее трудно обнаруживаемыми следует считать фатальные ошибки, связанные с прерыванием вычислительного процесса, поскольку любая такая ошибка, возникшая даже во второстепенном модуле, может прервать сложный вычислительный эксперимент, привести к зависанию программы или даже потере полученных результатов.

Природа таких ошибок, как правило, имеет «арифметический» характер, среди которых наиболее сложно идентифицируются ошибки типа деление на ноль (ошибки переполнения или потери порядка). Для вычислительных модулей (1) подобные ошибки возникают в точках разрыва второго рода функции и, следовательно, для их поиска можно применять методы глобального оптимизации [4, 5]:

$f(X) \to Min\,(Max), \quad X \in \Omega_f$.

1. Постановка задачи

В работе предлагается метод глобального поиска точек разрывов второго рода функции, идеи которого во многом заимствованы из метода многоэкстремальной оптимизации Р.Г. Стронгина [3].

Пусть областью поиска ошибок в вычислительном модуле является единичный квадрат $\Pi = [0,\ 1] \times [0,\ 1]$, который в процессе деления разбивается на четыре квадрата меньших размеров (см. рис. 1).

В узлах сформированной регулярной сетки вычисляются значения тестируемой функции z_{ij} . В качестве модели тестируемой функции рассмотрим многочлен четвертого порядка (второго порядка по каждой из переменных) вида

$$P_{2,2}(x, y) = a_0 x^2 y^2 + a_1 x^2 y + a_2 x y^2 + a_3 x^2 + a_4 y^2 + a_5 x y + a_6 x + a_7 y + a_8 . \qquad (2)$$

Коэффициенты многочлена (2) подбираются по результатам вычислений функции в узлах интерполяции z_{ij} .

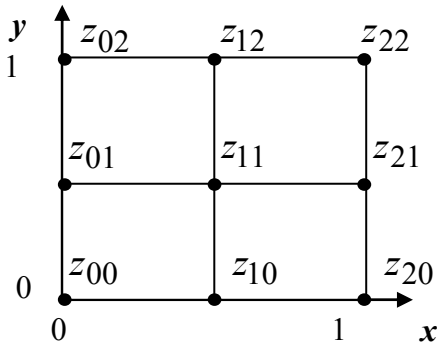

Рис. 1. Единичный квадрат

В соответствии с логикой метода характеризации [3] один из вновь образованных квадратов процедуры деления исходного квадрата подвергается очередному делению на квадраты меньшего размера и т.д. В общем случае происходит неравномерное разбиение области поиска.

2. Математическая модель поиска бесконечных разрывов функции

С практической точки зрения вместо многочлена (2) на регулярных сетках удобнее использовать двумерную первую интерполяционную формулу Ньютона.

Введем двойные разности высших порядков по следующей схеме. Определим частные конечные разности первого порядка

$$\begin{aligned} \Delta^1_x z_{ij} &= z_{i+1\,j} - z_{ij}, \\ \Delta^1_y z_{ij} &= z_{ij+1} - z_{ij}. \end{aligned} \qquad (3)$$

Повторно применяя эти операции, можно получить двойные разности высших порядков

$$\Delta^{n+m} z_{ij} = \Delta^{n+m}_{x^n y^m} z_{ij} = \Delta^n_{x^n} (\Delta^m_{y^m} z_{ij}) . \qquad (4)$$

Значения тестируемой функции в узлах интерполяционной сетки (см. рис. 1) представим матрицей

$$Z^T = \begin{Vmatrix} z_{00} & z_{10} & z_{20} \\ z_{01} & z_{11} & z_{21} \\ z_{02} & z_{12} & z_{22} \end{Vmatrix} . \qquad (5)$$

Для удобства вычислений введем новые переменные

$$q = \frac{x - x_0}{h_x},\ p = \frac{y - y_0}{h_y}. \qquad (6)$$

Полином $P_{2,2}(q, p)$ будем искать в виде

$$P_{2,2}(q, p) = A_0(p) + A_1(p) q + A_2(p) q(q-1) . \qquad (7)$$

При фиксированных значениях второй переменной ($p = 0, 1, 2$) построим одномерные полиномы в форме Ньютона.

Например, при p=0 многочлен может быть представлен следующим образом:

$$P^{(0)}(q) = z_{00} + \Delta_q z_{00} q + \frac{\Delta^2_{qq} z_{00}}{2} q(q-1). \tag{8}$$

Аналогично выводятся формулы для $p = 1$ и $p = 2$.

В формуле (7) коэффициент $A_0(p)$ описывает изменение первого члена многочлена (8) в зависимости от переменной p и может быть представлен в свою очередь многочленом второго порядка. Для его построения построим таблицу конечных разностей и получим:

$$A_0(p) = z_{00} + \Delta_p z_{00} p + \frac{\Delta^2_{pp} z_{00}}{2} p(p-1). \tag{9}$$

Аналогичным образом построим таблицу конечных разностей для коэффициентов $A_1(p)$ и $A_2(p)$. Коэффициенты $A_1(p)$ и $A_2(p)$ могут быть вычислены по формулам

$$A_1(p) = \Delta_q z_{00} + \Delta^{1+1}_{qp} z_{00} p + \frac{\Delta^{1+2}_{qpp} z_{00}}{2} p(p-1), \tag{10}$$

$$A_2(p) = \frac{1}{2}\Delta^2_{qq} z_{00} + \frac{1}{2}\Delta^{2+1}_{qqp} z_{00} p + \frac{\Delta^{2+2}_{qqpp} z_{00}}{4} p(p-1). \tag{11}$$

Введем матричную форму полного двухмерного полинома. Для этого определим матрицу конечных разностей

$$A = \begin{Vmatrix} z_{00} & \Delta_p z_{00} & \frac{1}{2}\Delta^2_p z_{00} \\ \Delta_q z_{00} & \Delta^{1+1}_{qp} z_{00} & \frac{1}{2}\Delta^{1+2}_{qpp} z_{00} \\ \frac{1}{2}\Delta^2_{qq} z_{00} & \frac{1}{2}\Delta^{2+1}_{qqp} z_{00} & \frac{1}{4}\Delta^{2+2}_{qqpp} z_{00} \end{Vmatrix}. \tag{12}$$

Если ввести вектора

$$Q = \begin{Vmatrix} 1 \\ q \\ q(q-1) \end{Vmatrix}, P = \begin{Vmatrix} 1 \\ p \\ p(p-1) \end{Vmatrix}, \tag{13}$$

тогда многочлен (7) с коэффициентами, вычисляемыми по формулам (9), (10) в векторной форме может быть представлен выражением

$$P_{2,2}(q, p) = Q^T AP = (AP, Q). \tag{14}$$

С помощью матричных форм легко подсчитать вторые частные производные для полиномиальной функции (14):

$$P''_{2,2\,qq}(q, p) = \frac{\partial^2 P_{2,2}}{\partial q^2} = \left(Q^T\right)''_{qq} AP = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}^T AP, \tag{15}$$

$$P''_{2,2\,pp}(q, p) = \frac{\partial^2 P_{2,2}}{\partial p^2} = Q^T AP''_{pp} = Q^T A \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}, \tag{16}$$

$$P''_{2,2\,qp}(q,p)=\frac{\partial^2 P_{2,2}}{\partial q\partial p}=\left(Q^T\right)'_q AP''_p=\begin{pmatrix}0\\1\\2q-1\end{pmatrix}^T A\begin{pmatrix}0\\1\\2p-1\end{pmatrix}. \quad (17)$$

Вторые частные производные от полиномиальной функции используются для вычисления характеристической функции, в которой одна из компонентов пропорциональна дифференциалу второго порядка.

С геометрической точки зрения второй дифференциал пропорционален главной части гауссовой кривизны поверхности. В частности в качестве приращений независимых переменных в первом приближении можно взять шаг частичной интерполяционной сетки разбиения очередного квадрата:

$dx=h_x$, $dy=h_y$, тогда

$$d^2P_{2,2}(x,y)=\frac{\partial^2 P}{\partial x^2}h_x^2+2\frac{\partial^2 P}{\partial x\partial y}h_xh_y+\frac{\partial^2 P}{\partial y^2}h_y^2. \quad (18)$$

Откуда

$$R(i)=\left(\left|\frac{\partial^2 P}{\partial x^2}\right|+2\left|\frac{\partial^2 P}{\partial x\partial y}\right|+\left|\frac{\partial^2 P}{\partial y^2}\right|\right)\left(h_xh_y\right)^r, \quad (19)$$

где r – параметр метода тестирования функции.

Если учесть, что

$$\frac{\partial^2 P(x,y)}{\partial x^2}=\frac{1}{h_x^2}\frac{\partial^2 P(q,p)}{\partial q^2}=\frac{1}{h^2}\frac{\partial^2 P(q,p)}{\partial q^2},$$

$$\frac{\partial^2 P(x,y)}{\partial y^2}=\frac{1}{h_y^2}\frac{\partial^2 P(q,p)}{\partial p^2}=\frac{1}{h^2}\frac{\partial^2 P(q,p)}{\partial p^2},$$

$$\frac{\partial^2 P(x,y)}{\partial x\partial y}=\frac{1}{h_xh_y}\frac{\partial^2 P(q,p)}{\partial q\partial p}=\frac{1}{h^2}\frac{\partial^2 P(q,p)}{\partial q\partial p},$$

то (18) приобретает вид

$$d^2P_{2,2}(x,y)=\frac{h^{2r}}{h^2}\left(\frac{\partial^2 P}{\partial x^2}+2\frac{\partial^2 P}{\partial x\partial y}+\frac{\partial^2 P}{\partial y^2}\right).$$

В результате

$$R(i)\approx\left|d^2P_{2,2}(x,y)\right|h^{2r-2}=\left(\left|\frac{\partial^2 P}{\partial x^2}\right|+2\left|\frac{\partial^2 P}{\partial x\partial y}\right|+\left|\frac{\partial^2 P}{\partial y^2}\right|\right)h^{2r-2}. \quad (20)$$

Смысл характеристики $R(i)$ заключается в том, что в окрестностях точек или линий разрыва второго рода параметр $R(i)\to\infty$, а в окрестностях локальных экстремумов он принимает наибольшие значения. Следовательно, $R(i)$ позволяет в процессе поиска вычислительных ошибок локализовать те места функции, где формируются ее бесконечные разрывы.

В процессе тестирования программного модуля формируется множество квадратов $D=\{D_1,D_2,...,D_n\}$, причем каждому квадрату ставится в соответствие векторный критерий $K(i)=(K_1(i),K_2(i))$.

Множество квадратов D_k упорядочивается лексикографически и из них выбирается квадрат, удовлетворяющий условию D_t : $(\max\limits_i K_1(i),\max\limits_i K_2(i))$. Выделенный квадрат D_t подвергается последующему делению. Процесс тестирования продолжается

до тех пор, пока не выполнится условие остановки $\|D_t\| < \varepsilon$, или не будет найдена точка разрыва.

3. Исследование производительности алгоритма поиска точек разрывов

Эффективность (производительность) предложенного алгоритма поиска точек разрывов второго рода будем оценивать по числу обращений к тестируемой функции. Объемы вычислений в десятки тысяч обращений к разрывной функции можно считать незначительными поскольку, как показали эксперименты, стандартные методы глобальной оптимизации либо вообще не находят разрывов, либо требуют существенно больших затрат машинного времени. Исследование производительности предлагаемого алгоритма проводилась с использованием нескольких модифицированных тестовых функций. Результаты испытаний представлены в табл. 1.

Таблица 1. Результаты испытаний

№	Тестируемая функция	Среднее число обращений к функции
1	**De Jong 2**. Овражная функция, один глобальный экстремум. Одна точка разрыва (X=(0.21;1.51) $F(x,y)=\frac{100}{100\ (x^2-y)+(1-x)^2+1}+\frac{1}{1-e^{-\frac{(x-0.21)^2+(y-1.51)^2}{0.01}}}.$ $-1.28 \le x, y \le 1.28.$	584
2	**Griewank.** Один глобальный и множество локальных максимумов, одна точка разрыва $F(x,y)=\frac{1}{\frac{x^2+y^2}{200}-\cos(x)\cos(y/\sqrt{2})+1},$ $-20 \le x, y \le 20.$	676
3	**Модифицированная Растригина**. Множество линий разрывов $F(x,y)=\frac{1}{(10\cos(2\pi x)-x^2)-(10\cos(2\pi y)-y^2)},$ $-5.12 \le x, y \le 5.12.$	181

Из таблицы видно, что в среднем точки разрыва функции второго рода идентифицируются предложенным алгоритмом за сравнительно небольшое количество обращений к функции. Сложнее решаются задачи, когда функция содержит «трубчатые» разрывы или имеет многочисленные локальные оптимумы. Значительно проще реализуется поиск линий разрывов тестируемой функции.

Заключение

Вычислительные эксперименты с тестовыми функциями показали, что методы глобальной оптимизации существенно сокращают трудоемкости алгоритмов тестирования вычислительных программных модулей.

Литература

1. Липаев В.В. Отладка сложных программ. Методы, средства, технология. М.: Энергатомиздат, 1993.
2. Коварцев А.Н. Автоматизация разработки и тестирования программных средств. Самара: Самар. гос. аэрокосм. ун-т. 1999.
3. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. М.: Наука, 1978.
4. Коварцев А.Н., Попова-Коварцева Д.А. Многомерный параллельный алгоритм глобальной оптимизации модифицированным методом половинных делений // В мире научных открытий. 2012. № 8.1(32).
5. Gergel V.P., Strongin R.G. Parallel computing for globally optimal decision making on cluster systems // Future Generation Computer Systems. 2005. V. 21. № 5.