ГУМАНИТАРНЫЕ НАУКИ

УДК 004.65

ОБ ИСПОЛЬЗОВАНИИ ТЕХНИКИ АВТОМАТИЧЕСКОГО ДИФФЕРЕНЦИРОВАНИЯ НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧИ ДВУМЕРНОЙ ФИЛЬТРАЦИИ

А. Р. Абдрахманов

aydarabd@yandex.ru

ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)

Аннотация. В работе рассмотрено три способа вычисления частных производных для матрицы Якоби в методах линеаризации Ньютоновского типа: аналитический или ручной способ, численное дифференцирование и с помощью техники автоматического дифференцирования (АД). Представлено сравнительное тестирование скорости работы некоторых доступных библиотек для АД на языке программирования C++. На примере полностью неявной численной схемы для численного решения задачи двумерной фильтрации нефти и воды представлены преимущества использования техники АД. Тестовые расчета верифицированы с гидродинамическим симулятором «РН-КИМ».

Ключевые слова: метод Ньютона; матрица Якоби; аналитическое дифференцирование; численное дифференцирование; автоматическое дифференцирование; многофазная фильтрация; симулятор РН-КИМ.

ВВЕДЕНИЕ

Большинство задач численного моделирования связано с решение нелинейных систем из алгебраических уравнений вида:

$$F(X) \equiv \begin{pmatrix} f_1(x_1, \dots, x_N) \\ \dots \\ f_N(x_1, \dots, x_N) \end{pmatrix} = 0, \qquad (1)$$

возникающих, например, при сеточной аппроксимации дифференциальных уравнений в частных производных. Общепринятым подходом для решения систем (1) является использование методов Ньютоновского типа:

$$X^{k+1} = X^k - J^{-1}(X^k) \cdot F(X^k), \qquad (2)$$

где k – номер итерации, $J^{-1}(X^k)$ – обратная матрица Якоби в векторе X^k.

Рассмотрим способы вычисления производных для матрицы $J^{-1}(X^k)$:

– Аналитическое дифференцирование;

– Численное дифференцирование;

– Автоматическое дифференцирование.

Аналитическое дифференцирование состоит в явном кодировании производных в программном коде. Данный метод является интуитивно понятным, однако обладает избыточностью программного кода и рядом ограничений. Если невязка $F(X)$ представлена как $F(X) \equiv R(Y(X))$, то вдобавок необходимо прописывать производную сложной функции $\frac{\partial R}{\partial X} = \frac{\partial R}{\partial Y}\frac{\partial Y}{\partial X}$. В дальнейшем любое изменение невязки влечет за собой «недешевые» каскадные модификации кода, подверженные риску возникновения ошибок.

Численное дифференцирование выполняется по определению:

$$f'(x) = \frac{f(x+\Delta x)-f(x)}{\Delta x} + O(\Delta x).$$

Данный способ избавляет от написания производных вручную, однако согласно (1) необходимо дополнительно посчитать $f_j(x_i + \Delta x),\ i = \overline{1,N}, j = \overline{1,N}$. Это может сильно сказаться на времени расчета в сторону замедления.

Техника АД устраняет недостатки первых двух способов. Для вычисления производных автоматически в некоторой точке x_0

программе достаточно лишь знать значение функции в точке $f(x_0)$ и ее производную в точке $f'(x_0)$. Затем перегрузить производные простых арифметических операций и производные элементарных функций.

Техника АД имеет ряд реализаций на C++. Были рассмотрены такие библиотеки как FADBAD [1], CPPAD [2] и ADEPT [3]. Для оценки скорости была взята функция $y = \sum_{i=0}^{n} x_i^i$ для $n = 10^6$. В результате тестирования выбрана самая производительная библиотека ADEPT (время расчета 1.4 сек, против 9.7 сек в CPPAD и 90.6 сек в FADBAD).

ПРИМЕНЕНИЕ ТЕХНИКИ АД ДЛЯ ДВУХФАЗНОЙ ДВУМЕРНОЙ ФИЛЬТРАЦИИ

Библиотека ADEPT верифицирована при решении задачи моделирования двухфазного потока жидкости в двумерном пространстве $[0; L] \times [0; L]$ [4]:

$$m\frac{\partial S}{\partial t} - \frac{k}{\mu_{\text{В}}}\left[\frac{\partial}{\partial x}\left(k_{\text{В}}(S)\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_{\text{В}}(S)\frac{\partial p}{\partial y}\right)\right] = 0, \quad (3)$$

$$-m\frac{\partial S}{\partial t} - \frac{k}{\mu_{\text{Н}}}\left[\frac{\partial}{\partial x}\left(k_{\text{Н}}(S)\frac{\partial p}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_{\text{Н}}(S)\frac{\partial p}{\partial y}\right)\right] = 0, \quad (4)$$

Здесь неизвестными являются:

$S = S(x, y, t)$ – водонасыщенность,

$p = p(x, y, t)$ – давление.

Дискретизируем уравнения (3) и (4) полностью неявной схемой [4]:

$$m\frac{S_{i,j}^{n+1} - S_{i,j}^{n}}{\tau} - \frac{k}{\mu_B}\frac{1}{h}\left[k_B\left(S_{i+\frac{1}{2},j}^{n+1}\right)\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{h} - \text{-kBSi-12,}\right.$$
$$\text{jn+1pi,jn+1-pi-1,jn+1h++kBSi,}$$
$$\left.\text{j+12n+1pi,j+1n+1-pi,jn+1h--kBSi,j-12npi,jn+1-pi,j-1n+1h=0,}\right. \quad (5)$$

$$-m\frac{S_{i,j}^{n+1} - S_{i,j}^{n}}{\tau} - \frac{k}{\mu_{\text{Н}}}\frac{1}{h}\left[k_{\text{Н}}\left(S_{i+\frac{1}{2},j}^{n+1}\right)\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{h} - k_{\text{Н}}\left(S_{i-\frac{1}{2},j}^{n+1}\right)\frac{p_{i,j}^{n+1} - p_{i-1,j}^{n+1}}{h} + k_{\text{Н}}\left(S_{i,j+\frac{1}{2}}^{n+1}\right)\frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{h} - k_{\text{Н}}\left(S_{i,j-\frac{1}{2}}^{n}\right)\frac{p_{i,j}^{n+1} - p_{i,j-1}^{n+1}}{h}\right] = 0, \quad (6)$$

На рис. 1 представлена функция для задания вектора невязок и функция для вычисления матрицы Якоби через библиотеку ADEPT:

```
void BL::Nev9zka1(vector<vector<adouble>>& F2d,                  // вектор невязок
                  vector<vector<adouble>>& X2d_new_for_P,        // вектор давлений
                  vector<vector<adouble>>& X2d_S,                // вектор насыщенностей
                  vector<vector<adouble>>& Uold_for_S, int n) {

    for (int i = 1; i < n - 1; i++) {

        for (int j = 1; j < n - 1; j++) {

            F2d[i][j] = - poro_ * (X2d_S[i][j] - Uold_for_S[i][j]) / tau_ - (kabs_ / muo_) / h_ * (
                Kro(Liplus12(X2d_S, i, j)) * (X2d_new_for_P[i + 1][j] - X2d_new_for_P[i][j]) / h_ -
                Kro(Liminus12(X2d_S, i, j)) * (X2d_new_for_P[i][j] - X2d_new_for_P[i - 1][j]) / h_ +
                Kro(Ljplus12(X2d_S, i, j)) * (X2d_new_for_P[i][j + 1] - X2d_new_for_P[i][j]) / h_ -
                Kro(Ljminus12(X2d_S, i, j)) * (X2d_new_for_P[i][j] - X2d_new_for_P[i][j - 1]) / h_
                );
        }
    }
}
void Base::GetJacobian(adept::Stack &stack, vector<adouble> &F, vector<adouble> &X,
            vector<vector<double>> &Jacobian, int n) {

    vector <double> jac(n*n);   // вектор для получения матрицы якоби
    for (int i = 0; i < n; i++)
        F[i].set_gradient(1.0); // определение вектора в качестве целевой функции

    stack.compute_adjoint();  // запуск записи в стэк
    stack.independent(&X[0], n); // определение вектора независимых переменных
    stack.dependent(&F[0], n);  // определение вектора невязок
    stack.jacobian(&jac[0]); // вычисление матрицы якоби
    int k = 0;

    // запись матрицы якоби в двумерный вектор из одномерного
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; ++j) {
            Jacobian[j][i] = jac[k++];
        }
    }
}
```

Рис. 1. Тело функции для вычисления матрицы Якоби через инструмент АД

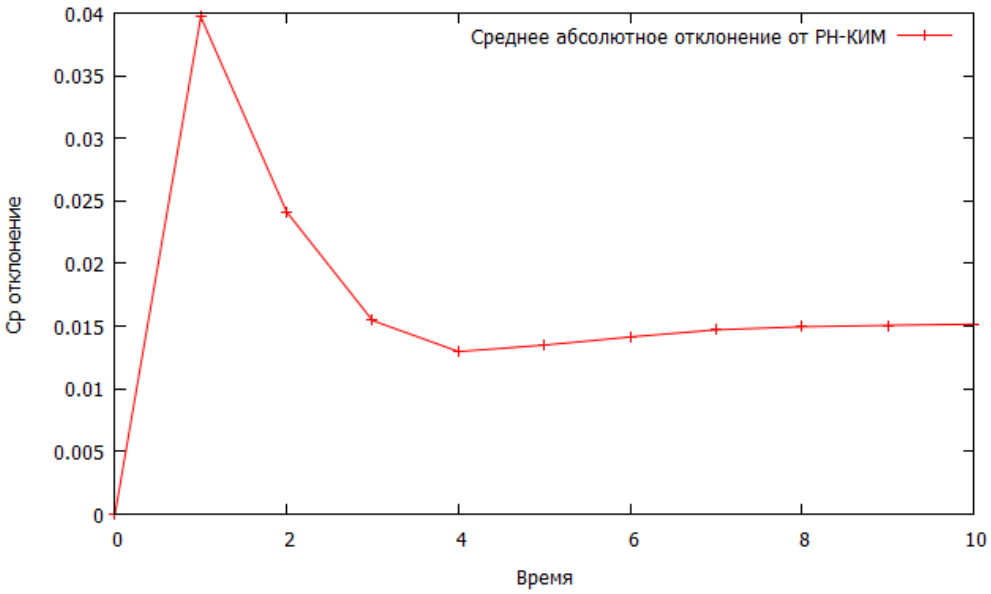

Рис. 2. Среднее абсолютное отклонение от результатов в РН-КИМ

В качестве проверки адекватности численного решения был проведен ряд тестов, рассчитанных с помощью коммерческого симулятора «РН-КИМ» [5].

На рис. 2 представлен график среднего абсолютного отклонения от результатов РН-КИМ для одного из тестов (размеры модели 15×15 метров, шаг по сетке 1 метр, время моделирования 5 часов с шагом по времени 30 минут).

ЗАКЛЮЧЕНИЕ

В данной работе было продемонстрировано применение техники автоматического дифференцирования для вычисления матрицы Якоби для метода Ньютона. Проведена верификация результатов использования библиотеки ADEPT на примере решения задачи двумерной фильтрации в системе (нефть+вода) с коммерческим гидродинамическим симулятором РН-КИМ.

СПИСОК ЛИТЕРАТУРЫ

1. **Ole S. and Claus B.** Flexible AD using templates and operator overloading in C++ [Электронный ресурс]. URL: http://www.fadbad.com/download/FlexibleAD-talk.pdf (дата обращения 02.02.2020). **[** S. Ole and B. Claus (2020, Feb. 2). *Flexible AD using templates and operator overloading in C++* [Online]. Available: http://www.fadbad.com/download/FlexibleAD-talk.pdf**]**

2. **Bradley M. B.** A C++ Algorithmic Differentiation Package [Электронный ресурс]. URL: https://coin-or.github.io/CppAD/ doc/get_started.cpp.htm (дата обращения 02.02.2020). **[** M. B. Bradley (2020, Feb. 2). *A C++ Algorithmic Differentiation Package* [Online]. Available: https://coin-or.github.io/ CppAD/doc/get_started.cpp.htm**]**

3. **Robin J. H.** Adept C++ Software Library: User Guide [Электронный ресурс]. URL: http://www.met.reading.ac.uk/clouds/adept/adept_documentation_2.0.pdf (дата обращения 02.02.2020). **[** H. J. Robin (2020, Feb. 2). *Adept C++ Software Library: User Guide* [Online]. Available: http://www.met.reading.ac.uk/clouds/adept/adept_documentation_2.0.pdf**]**

4. **Азиз, Х., Сеттари Э.** Математическое моделирование пластовых систем - Москва-Ижевск: Институт компьютерных исследований, 2004. – 416 с [K. Aziz and A. Settari, *Petroleum reservoir simulation*, Moscow-Izhevsk: Institute for Computer Research, 2004]

5. **ООО «РН-БашНИПИнефть»** Гидродинамический симулятор залежей углеводородов «РН-КИМ», URL: https://rn.digital/rnkim/ . [*Reservoir simulator RN-KIM.* Available: https://rn.digital/en/rnkim/]

ОБ АВТОРЕ

АБДРАХМАНОВ Айдар Радикович, магистрант 1-го курса кафедры математики, общенаучного факультета, УГАТУ.

METADATA

Title: About using the automatic differentiation techniques based on the solution of the two-dimentional filtering problem

Author: A. R. Abdrakhmanov

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

Email: aydarabd@yandex.ru

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), no. 2 (23), pp. 147-150, 2020. ISSN 2225-9309 (Print).

Abstract: In this work considered three ways computing partial derivatives for Jacobi matrix in the Newton's methods

such as analytical differentiation, numerical differentiation and with automatic differentiation techniques (AD). There is shown comparing speed test of the several AD libraries on the programming language C++. On the example fully implicit scheme for the numerical solution of the two-dimentional filtering problem there is shown benefits of using AD techniques. Test results were verified with the reservoir simulator RN-KIM.

Key words: Newton's method; Jacobi matrix; analytical differentiation; numerical differentiation; automatic differentiation; multiphase filtration; reservoir simulator RN-KIM.

About author:

ABDRAKHMANOV, Aydar Radikovich, Master student 1 year, Ufa state aviation technical University

ОПЕЧАТКИ

Стр.	Напечатано	Следует читать
138 (название статьи на английском)	**Title:** Web OLAP conceptual data model design on the basis of situation-oriented database	**Title:** An algorithm of Raman spectroscopy data processing to generate a training dataset for an artificial neural network in the differential diagnosis of malignant diseases
148 (формула (5))	$m\frac{S_{i,j}^{n+1}-S_{i,j}^{n}}{\tau}-$ $-\frac{k}{\mu_B}\frac{1}{h}\left[k_B\left(S_{i+\frac{1}{2},j}^{n+1}\right)\frac{p_{i+1,j}^{n+1}-p_{i,j}^{n+1}}{h}-\right.$ $-kBSi-12,$ $jn+1pi,jn+1-pi-1,jn+1h++kBSi,$ $j+12n+1pi,j+1n+1-pi,jn+1h--kBSi$ $,j-12npi,jn+1-pi,j-1n+1h=0,$	$m\frac{S_{i,j}^{n+1}-S_{i,j}^{n}}{\tau}-$ $-\frac{k}{\mu_B}\frac{1}{h}\left[k_B\left(S_{i+\frac{1}{2},j}^{n+1}\right)\frac{p_{i+1,j}^{n+1}-p_{i,j}^{n+1}}{h}-\right.$ $-k_B\left(S_{i-\frac{1}{2},j}^{n+1}\right)\frac{p_{i,j}^{n+1}-p_{i-1,j}^{n+1}}{h}+$ $+k_B\left(S_{i,j+\frac{1}{2}}^{n+1}\right)\frac{p_{i,j+1}^{n+1}-p_{i,j}^{n+1}}{h}-$ $\left.-k_B\left(S_{i,j-\frac{1}{2}}^{n}\right)\frac{p_{i,j}^{n+1}-p_{i,j-1}^{n+1}}{h}\right]=0,$
147-150 (статья Абдрахманова А.Р. «Об использовании техники автоматического дифференцирования на примере решения задачи двумерной фильтрации»)	отнесена к разделу «Гуманитарные науки»	относится к разделу «Технические науки»