

Прикладная математика и информатика

Определение временной и емкостной сложности алгоритмов при равномерном весовом критерии

В работе исследуется такая проблема, как емкостная сложность алгоритмов, характеризующая длины описания алгоритмов.

Сложность алгоритма может быть различной и при фиксированном размере входа, но различных входных данных. Примером может послужить поиск контрарной пары двух дизъюнктов фиксированной длины, контрарные литералы могут оказаться первыми или последними в этих дизъюнктах. Следовательно, время поиска контрарной пары в этих случаях окажется различным, поэтому различают сложность в худшем и в среднем случае.

Сложность в худшем случае – это максимальная при данном размере входа сложность.

Средняя сложность – это средняя по всем входам данного размера сложность.

Емкостная сложность в худшем случае – это функция $S(n)$ – равная максимальной (по всем входам размера n) из сумм емкостей всех ячеек (элементов) памяти к которым было обращение.

Для точного определения емкостной сложности требуется знать объем памяти используемой каждой переменной. Так как это значение зависит от конкретной вычислительной машины, в теории сложности алгоритмов принято оценивать порядок сложности.

При определении порядка емкостной сложности в качестве единицы необходимо выбрать некоторый элементарный в данном классе алгоритмов объект, таким объектом в задаче логического вывода на основе метода резолюций является литерал.

При оценке сложности могут использоваться различные критерии:

1. Равномерный весовой критерий предполагает, что каждая переменная в алгоритме занимает одну ячейку памяти независимо от принимаемых ей значений.

2. Логарифмический весовой критерий учитывает ограниченность размера реальной ячейки памяти.

Например, необходимо определить временную и емкостную сложность алгоритма сортировки массива по возрастанию методом «пузырька» при равномерном весовом критерии. В соответствии с данным методом для всех элементов массива, начиная со второго, последовательно выполняется «всплывание», при котором текущий элемент, если он меньше предыдущего, меняется с ним местами. Обозначим i – текущий номер элемента при просмотре массива в глубину, j – номер текущего элемента при «всплывании» (процедура `elem_up`), $A(j)$ – j -ый элемент массива, n – число элементов в массиве.

Определим временную сложность при равномерном весовом критерии. Очевидно, что размером входа для данного алгоритма следует считать число n элементов в массиве, а шагом алгоритма – тело цикла в процедуре «всплывание». В теле основной программы имеем $(n-1)$ проходов по циклу. В процедуре `elem_up` («всплывание» элемента) – в худшем случае выполняется i проходов по циклу. Тогда, общее число шагов в алгоритме равно:

$$\sum_{i=2}^n i = (n+2) * (n-1) / 2 = (n^2 + n - 2) / 2$$

Например, при $n = 4$ осуществляется три входа в основной цикл $(n-1)$ и 6 операций в подпрограмме $(n+2)$.

Таким образом, временная сложность алгоритма при равномерном весовом критерии есть $O(n^2)$.

Емкостная сложность при равномерном весовом критерии, очевидно, определяется размером n массива, поэтому есть $O(n)$.

Литература

1. Архангельский Язык Pascal и основы программирования в Delphi. Учебное пособие - М.: ООО «Биноп-Пресс», 2004 – 496с.
2. Хомоненко А., Гофман В., Никифоров В. Delphi 7. – СПб.: БХВ- Петербург, 2004. – 1200с.

Приложение определенного интеграла к нахождению работы подъемного крана

Задача: Какую работу затрачивает подъемный кран при извлечении железобетонной надолбы со дна реки глубиной 5 м, если надолба имеет форму правильного тетраэдра с ребром 1 м, а плотность железобетона 2500 кг/м³?

Решение:

Высота тетраэдра $h = \sqrt{6} / 3$ м, объем тетраэдра $V = \sqrt{2} / 12$ м³. Вес надолбы в воде

$$P = (1/12) \cdot \sqrt{2} \cdot 2500 \cdot 9,8 - (1/12) \cdot \sqrt{2} \cdot 1000 \cdot 9,8 = 1225\sqrt{2} \text{ Н,}$$

Поэтому работа на извлечение надолбы до момента появления на поверхности воды ее вершины составляет

$$A = 1225\sqrt{2}(5 - h) = 1225\sqrt{2}(5 - \sqrt{6}/3) \cong 7227,5 \text{ Дж}$$

Теперь найдем работу A_1 на извлечение надолбы из воды. Пусть вершина тетраэдра вышла на высоту $5+y$, тогда объем малого тетраэдра, вышедшего из воды равен $3\sqrt{3}y^3 / 8$, а вес тетраэдра

$$P(y) = \frac{2500 \cdot 9,8}{12} \sqrt{2} - \left(\frac{1}{12} \sqrt{2} - \frac{1}{8} y^3 3\sqrt{3} \right) 1000 \cdot 9,8 \text{ Н.}$$

Следовательно,

$$\begin{aligned} A &= \int_0^h \left(\frac{24500}{12} \sqrt{2} - \frac{9800}{12} \sqrt{2} + \frac{9800}{8} 3y^3 \sqrt{3} \right) dy = \\ &= \int_0^{\sqrt{6}/3} (1225\sqrt{2} - 3675\sqrt{3}y^3) dy = \left[1225\sqrt{2}y + \frac{3675}{4} \sqrt{3}y^4 \right]_0^{\sqrt{6}/3} \approx 2082,5 \text{ Дж.} \end{aligned}$$

Отсюда

$$A = A_0 + A_1 = 7227,5 + 2082,5 = 9130 \text{ Дж} = 9,31 \text{ кДж.}$$

Приложения определенного интеграла к нахождению давления воды

Задача:

Водопроводная труба имеет диаметр 6 см; один конец ее соединен с баком, в котором уровень воды на 100% выше верхнего края трубы, а другой закрыт заслонкой. Найти полное давление на заслонку.

Решение:

Заслонка представляет собой круг радиуса 3 см. Разобьем площадь этого круга на элементы – полосы, параллельные поверхности воды. Площадь одного такого элемента, находящегося на расстоянии y от центра, равна (с точностью до бесконечно малых высшего порядка)

$$dS = 2\sqrt{9 - y^2} dy \text{ см}^2.$$

Найдем силу давления, испытываемую этим элементом.

$$dP = 2\rho g(103 - y)\sqrt{9 - y^2} dy = 1960(103 - y)\sqrt{9 - y^2} dy \text{ дин}$$

Здесь $\rho = 1 \text{ г/см}^3$. следовательно.

$$P = 1960 \int_{-3}^3 (103 - y)\sqrt{9 - y^2} dy = 1960 \left[103 \left(\frac{y}{2} \sqrt{9 - y^2} + \frac{9}{2} \arcsin \frac{y}{3} \right) + \frac{1}{3} (9 - y^2)^{3/2} \right]_{-3}^3$$

$$= 980 \cdot 927\pi = 908460\pi \text{ эрг} \approx 0,09\pi \text{ Дж.}$$

Задача:

Найти величину давления воды на вертикальную стенку в форме полукруга, диаметр которого 6 м и находится на поверхности воды. Плотность воды $\rho = 1000 \text{ кг/м}^3$.

Решение:

Дифференциал давления на элементарную площадку выразится так:

$$dP = 2\rho g x \sqrt{9 - x^2} dx = 19600x\sqrt{9 - x^2} dx$$

Отсюда

$$P = 19600 \int_0^3 x\sqrt{9 - x^2} dx = -\frac{19600}{3} (9 - x^2)^{3/2} \Big|_0^3 = 176400 \text{ Н} = 176,4 \text{ кН.}$$

Задача:

Какое давление испытывает прямоугольная пластинка длиной a и шириной b ($a > b$), если она наклонена к горизонту жидкости под углом α и ее большая сторона находится на глубине h ?

Решение:

Площадь выделенной на глубине x элементарной полоски равна

$$dS = (a / \sin \alpha) dx$$

Следовательно, элемент давления $dP = (ax\rho g / \sin \alpha) dx$

ρ – плотность жидкости.

Отсюда находим

$$P = a\rho g \int_h^{h+b \sin \alpha} \frac{x dx}{\sin \alpha} = \frac{a\rho g}{\sin \alpha} \cdot \frac{1}{2} x^2 \Big|_h^{h+b \sin \alpha} = \frac{a\rho g}{2 \sin \alpha} [(h^2 + 2bh \sin \alpha + b^2 \sin^2 \alpha) - h^2] =$$

$$= ab\rho g \left(h + \frac{1}{2} b \sin \alpha \right)$$

И.Р. Сарибжанов
 Научный руководитель: профессор, д-р техн. наук. В.В. Чекушкин
 Муромский институт Владимирского государственного университета
 602264 г. Муром, Владимирской обл., ул. Орловская, д. 23
 e-mail: oid@mivlgu.ru

Ускоренный алгоритм деления чисел

Требуется разработать быстродействующий алгоритм вычисления значения функции деления чисел в формате с фиксированной точкой. Значения чисел лежат в диапазоне $x \in [0,1]$.

Предлагается следующий метод. Операция деления заменяется операцией умножения числителя на обратную величину знаменателя: $A/B = A \cdot (1/B)$. Следовательно, задача сводится к поиску обратной величины знаменателя.

Поскольку формат представления чисел имеет вид $x=0.xxxxxxx$, то возникает проблема, что в окрестности нуля функция устремляется к бесконечности и аппроксимация затруднена.

Используем следующий подход: поскольку аппроксимация происходит на интервале $x \in [0,1]$, весь интервал делим на некоторое количество отрезков и аппроксимацию производим для каждого отрезка отдельно, тем самым будет уменьшаться погрешность аппроксимации на интервале.

Прямое применение полиномов не дает высоких точностных результатов при поиске обратной величины знаменателя, поэтому предлагается использовать итерационный метод Ньютона, где каждое следующее приближение вычисляется из предыдущего, применением формулы:

$$r_{i+1} = r_i(2 - xr_i)$$

Но итерационные также имеют недостатки, они не всегда сходятся, т.е. не обязательно возвращают искомую величину. Поэтому в качестве начального приближения возьмем значение полинома 2й степени, это обеспечит скорейшую сходимость.

Чем ближе к нулю значение делителя, тем большее количество итераций необходимо провести для достижения точности.

В таблице 1 приведены результаты

интервал	Начальное приближение	\sum оп	Погрешность (дм)
[0.2,1]	$6.80965 - x(13.40626 + 7.76287x)$	1 0	$8.76 \cdot 10^{-4}$
[0.025,0.2]	$40.47461 - x(446.27948 + 1384.97597x)$	1 3	$6.001 \cdot 10^{-4}$
[0.0038,0.025]	$296.49793 - x(24598.9201 + 584352.24847x)$	1 3	$5.838 \cdot 10^{-4}$

\sum оп – количество операций для вычисления значения. дм – значение максимальной абсолютной погрешности.

При таком подходе, вычисление значения обратной величины делителя при $x < 0.0038$ требует деление интервала на более мелкие отрезки, и вычисление полиномов для этих интервалов. Следует признать, что этот подход неэффективным.

Для решения задачи предлагается иной подход: будем преобразовывать знаменатель к формату с ненулевой первой цифрой после точки. То есть домножать оба операнда на степень 10, например, если значение операндов равно, 0.00058 и 0.0000185, то выражение будет иметь вид

$$\frac{0.00001855}{0.00058} = \frac{0.01855}{0.58} = 0.01855 \times \frac{1}{0.58}$$

Соответственно, при таком преобразовании операндов значение знаменателя будет принадлежать диапазону $x \in [0.1..1]$, следовательно, и аппроксимацию функции поиска обратной

величины необходимо производить в этом диапазоне.

Получение значения функции будем производить по тому же пути что и в первом случае – комбинированный метод, за исключением того, что теперь рассмотрим сразу весь интервал целиком.

В качестве начального приближения возьмем полином второй степени

$$8.94022 - x(20.91834 + 13.40057x)$$

При трех итерациях по методу Ньютона, значение максимальной погрешности на интервале $x \in [0.1..1]$ составляет $\text{дм} = 1,014 * 10^{-3}$. Вычислительные затраты составляют 13 операций, при этом количество констант, хранимых в памяти, равно всего 3, тогда как в первом алгоритме для каждого интервала хранится свой набор констант из трех констант. Если количество итераций увеличить до 4, то потребуется выполнить 16 операций и максимальное значение погрешности сократиться до $\text{дм} = 1,029 * 10^{-6}$.

Литература

1. Чекушкин В.В. Вычислительные процессы в информационно-измерительных системах / В.В. Чекушкин, В.В. Булкин. - Муром: Изд.- полиграфический центр МИ ВлГУ, 2009.-120с.

Улучшение алгоритмов вычисления функции $\arctg(x)$

Функция арктангенса имеет важное значение при преобразовании ортогональных составляющих сигнала в фазу, также при преобразовании прямоугольных координат в полярные.

Арктангенс a есть угол, заключенный в интервале от $-\pi/2$ до $\pi/2$ (или от -90° до $+90^\circ$), тангенс которого равен a . При этом угол, например, азимут ψ , изменяется в пределах $\psi \in [0^\circ, 360^\circ]$, но достаточно вычислить значение арктангенса в пределах $\psi \in [-90^\circ, 90^\circ]$ (Рис. 1.а).

Будем аппроксимировать только правую часть функции, т.е. половину, где аргумент функции больше нуля. В качестве показателя точности выберем максимальное значение абсолютной погрешности $d_m = L(x) - Z_n(x)$

При аппроксимации функции арктангенса применяется полином Тейлора:

$$\arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} \quad (1)$$

Применение многочлена Тейлора эффективно для значений аргумента на интервале $x \in [-1, 1]$, т.е. для значений функции $[-45^\circ, 45^\circ]$. Рассмотрим полином Тейлора при различном числе членов полинома (Таблица. 1).

Таблица 1. Полиномы Тейлора для аппроксимации функции $\arctg(x)$ на интервале $x \in [-1, 1]$

№	количество членов	максимальная погрешность, d_m
1	1	0,2146
2	2	0,1187
3	3	0,0813
4	4	0,0616
5	5	0,0495
6	6	0,0414
9	9	0,0277
1	20	0,0125

Повышение степени полинома увеличивает количество вычислительных операций, но не обеспечивает эффективное снижение погрешности. Наиболее эффективным стоит считать полином из 5 членов, поскольку достигается невысокая погрешность при умеренных вычислительных затратах.

$$Z1(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} \quad (2)$$

Для уменьшения погрешности вычисления используем имеющий лучшие потенциальные возможности аппроксимации функции полином наилучшего приближения. (Рис. 1.а).

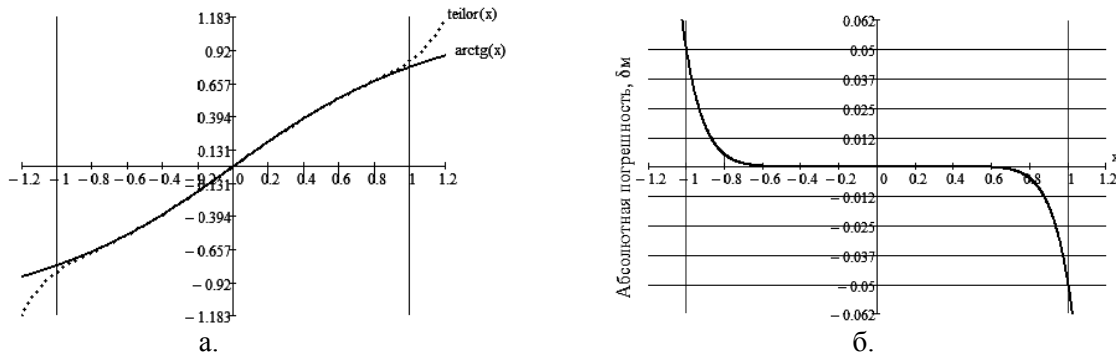


Рис.1.а. Графики функций $teilor(x)$ – аппроксимирующая функция (2), $arctg(x)$ – функции арктангенса. **б.** абсолютная погрешность полнима Тейлора.

Наряду с полиномом Тейлора известен и другой алгоритм [1]:

$$Z2(x) = (((k9 * x^2 + k7)x^2 + k5)x^2 + k3)x^2 + k1)x \quad (3)$$

где $k1 = 0.99987$, $k3 = -0.33038$, $k5 = 0.1855$, $k7 = -0.08583$, $k9 = 0.02121$;

Данный алгоритм так же эффективно аппроксимирует функцию на интервале $x \in [-1, 1]$. Проведем моделирование вычислительного процесса в этих пределах (Рис.2.а).

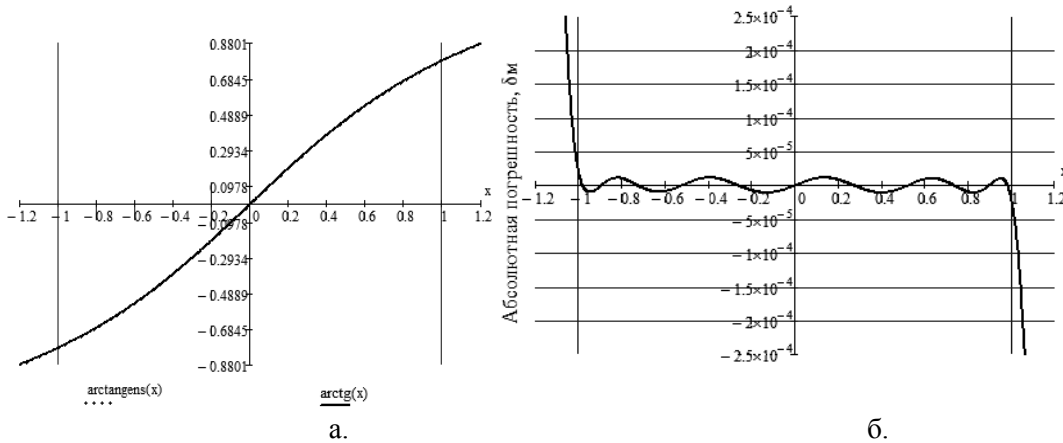


Рис.2 а. Графики функций $arctangens(x)$ – аппроксимирующая функция (2), $arctg(x)$ – функции арктангенса. **б.** абсолютная погрешность функции (3).

Максимальная погрешность на заданном интервале составляет $d_M = 2,18 * 10^{-5}$.

Сравнивая функции (2) и (3) следует, что аппроксимирующая функция (3) является оптимальной, поскольку при меньшем числе вычислительных операций, имеет меньшую погрешность.

Предлагается следующая функция:

$$Z3(x) = 0.9714x - 0.189x^3 \quad (4)$$

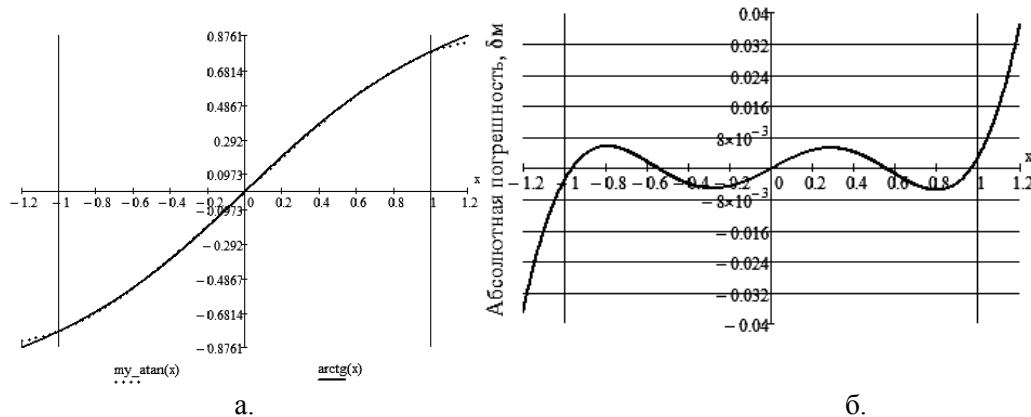


Рис.3 а. Графики функций $my_atan(x)$ – аппроксимирующая функция (2), $arctg(x)$ – функции арктангенса. **б.** абсолютная погрешность функции (4).

Алгоритм обеспечивает погрешность метода $d_M = 5,614 * 10^{-3}$.

Из таблицы (1) и Рис.3.б выражения (4) следует, что обеспеченно при использовании полинома третьей степени при одинаковых аппаратурных затратах снижение погрешности в $0.1187/5.614 \cdot 10^{-3} = 21$ раз.

Литература

1. Чекушкин В.В. Вычислительные процессы в информационно-измерительных системах/ В.В. Чекушкин, В.В. Булкин. - Муром: Изд.- полиграфический центр МИ ВлГУ, 2009.-120с.

Разработка быстродействующих алгоритмов вычисления значения функции $tg(x)$

Функция тангенса считается одной из важнейших функций. Она применяется во многих областях науки, в том числе является угловым коэффициентом касательной прямой к функции. Необходимо аппроксимировать функцию $Z = tg(x)$ (1) на интервале $x \in [0; p/2]$.

Существует алгоритм нахождения приближенного значения тангенса на основе полинома Маклорена.

$$tg(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \dots = \sum_{n=1}^{\infty} \frac{B_{2n}(-4)^n(1-4^n)}{(2n)!} x^{2n-1} \quad (2)$$

для всех $|x| < \frac{\pi}{2}$ где B_{2n} — Числа Бернулли.

Разложив ряд до седьмой степени, дает выражение.

$$Z_1(x) = x + \frac{x^3}{3} + \frac{2x^5}{5} + \frac{17x^7}{315} \quad (3)$$

Для обеспечения наиболее высокой эффективности представления результатов при реализации метода проведем предварительное моделирование вычислительного процесса, построив оба графика функций на одной координатной плоскости (Рис. 1. а)

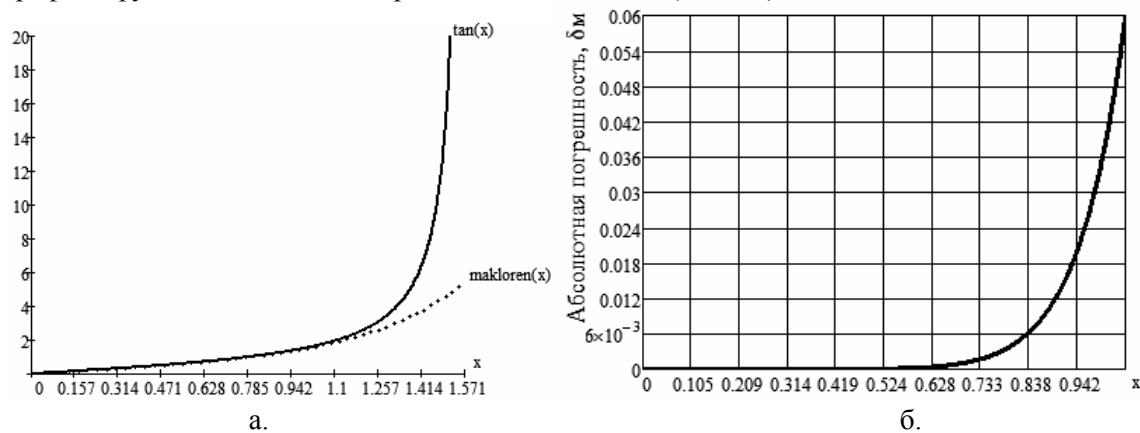


Рис.1. а. Графики функций (1) – $\tan(x)$ и приближенной функции (3) – $\text{makloren}(x)$

б. Абсолютное отклонение функции (3) на интервале $x \in (0; p/3)$.

Как видно из графика относительно погрешности (Рис. 1.б) при необходимой погрешности менее 0.06 данный метод эффективен только на отрезке $x \in (0, p/3)$, так как максимальная погрешность в точке $x = p/3$ составляет $\text{дм} = 0.059$. Если значение угла имеет большую величину, погрешность стремительно возрастает. При значении угла $x = p/4$ относительная погрешность составляет $\text{дм} = 3.32 \times 10^{-3}$.

Данный метод имеет серьезный недостаток, выраженный в количестве вычислительных операций.

Рассмотрим следующий алгоритм нахождения приближенного значения $Z = tg(x)$. Существует алгоритм, дающий приближенное значение функции $Z = tg(x)$ через отношение:

$$Z_2(x) = \frac{x(A * g + 1.0)}{(B * g + C) * g + 1.0} \quad (4)$$

где $A = -0.0958017723$, $B = -0.00971685835$, $C = -0.429135777$, $g = x^2$.

Для дальнейшего анализа отобразим оба графика функций $Z(x) = tg(x)$ и $Z_2(x)$ на одной координатной плоскости (Рис.2.а).

Для анализа точности функции (4) необходимо построить график погрешности (Рис.2.б).

Для сравнения данной функции (4) с формулой, полученной при использовании ряда Маклорена (3), вычислим погрешности при тех же значениях аргумента x .

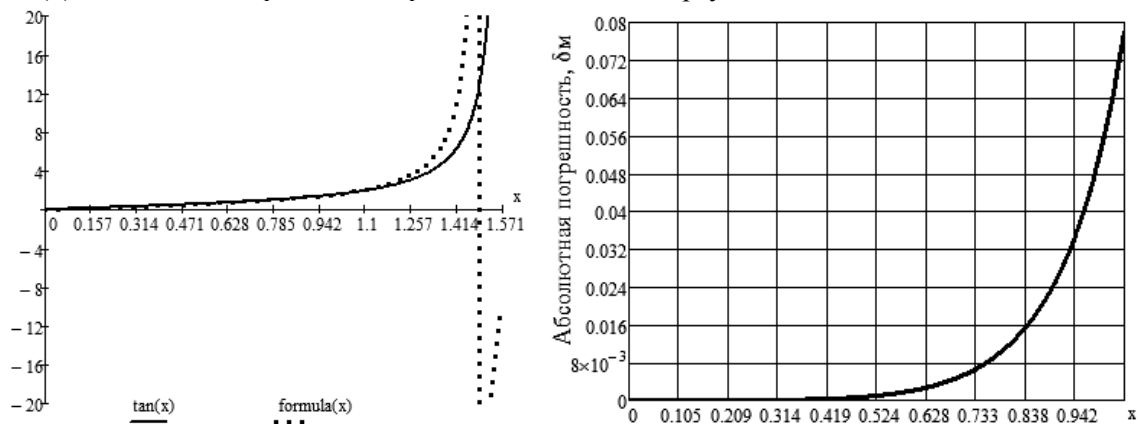


Рис.2. а. График функций (1) – $\tan(x)$ и приближенной функции (4) – $\text{formula}(x)$

б. Абсолютное отклонение функции (4) на интервале $x \in (0, \pi/3)$.

Поскольку график погрешности восходящий на всем промежутке, следовательно, вычисленные значения функции (4) в любой точке гарантирует, что все значения погрешности при значении аргумента меньшем данной величины будут ниже полученной величины погрешности. Таким образом, при значении аргумента $x = \pi/3$ функция (4) имеет значение погрешности равное $\text{дм} = 0.08$, а при значении аргумента $x = \pi/4$ погрешность равно $\text{дм} = 0.01$.

Сравнивая значения погрешностей с предыдущим методом видно, что второй метод имеет вычислительные показатели ниже, что выражается в погрешности, к тому же данный метод в своей формуле содержит трудоемкую операцию деления, что влечет за собой дополнительные сложности вычисления.

Для аппроксимации функции тангенса предлагается использовать полиномы наилучшего приближения. Поскольку вышеописанные методы эффективны в пределах $x \in [0; 60^\circ]$, вычислим полином наилучшего приближения 4го порядка для этого интервала.

$$Z_3 = x(0.89265 * x(0.80695 + x(-1.4455 + 1.30747x))) \quad (5)$$

Выражение (5) обеспечивает абсолютную погрешность на заданном интервале не более $\text{дм} = 4.34 \times 10^{-3}$.

Если уменьшить интервал до $x \in [0; 45^\circ]$, то получаем полином наилучшего приближения (6), который обеспечивает уменьшение значение максимального еще на порядок, и максимальная погрешность составляет $\text{дм} = 3.72 \times 10^{-4}$ (Рис.3.б).

$$Z_3 = x(0.98698 * x(0.13944 + x(-0.11363 + 0.5095x))) \quad (6)$$

В результате было установлено, что для аппроксимации функции тангенса в интервале $x \in [0, \pi/3]$ наиболее эффективным методом является использование полиномов наилучшего приближения 4й степени. При невысоких вычислительных затратах обеспечивается высокая точность в сравнении с другими алгоритмами.

Литература

1. Чекушкин В.В. Вычислительные процессы в информационно-измерительных системах/ В.В. Чекушкин, В.В. Булкин. - Муром: Изд.- полиграфический центр МИ ВлГУ, 2009.-120с.

Применение линейного программирования для усовершенствования конструкции инструмента

Покажем на примере исследования протяжек возможность и целесообразность применения одного из методов операционного анализа – линейного программирования, которое наиболее просто обеспечивает оптимизацию и позволяет учесть любое сколь угодно большое число ограничений, накладываемых на конструктивные параметры инструмента.

Основными параметрами протяжек являются подъем a_z , шаг t , высота h зубьев и число зубьев в группе z_c . За критерий оптимизации конструкции протяжки примем ее наименьшую возможную длину. С уменьшением длины снижается стоимость инструмента вследствие экономии дорогостоящего инструментального материала и повышается производительность протягивания. Длина режущей части протяжки зависит от угла θ наклона образующей конуса, на котором располагаются режущие кромки. Чем больше θ , тем короче протяжка. Это условие можно записать целевой функцией оптимизации, определяющей тангенс угла: максимизировать целевую функцию $\Phi = a_z : t \rightarrow \max$ при следующих ограничениях: $p \leq Q$; $p \leq [\sigma_x] F_x$; $p \leq \pi [\sigma_1] (r_1 - h)^2$; $p = \pi d C_p a_z^\lambda z_{\max} K_\gamma K_c K_u$; $\pi h^2 / (4a_l) \geq K_{\min} l/t \geq (z_{\min})$; $l/t \leq (z_{\max})$; $t \geq t_{\min}$; $h \leq 0.17d$; $a_z \geq a_{z \min}$; $t \geq 2.5h$.

Преобразуем математическую модель протяжки в линейный вид, логарифмируя целевую функцию и неравенства-ограничения. Для этого заменим выражения $z_{\max} = \frac{l_2}{t} + 1$ и $r_1 - h$ степенными зависимостями, найденными методом наименьших квадратов: $z_{\max} = 1,7l_2^{0.79} t^{-0.79}$; $(r_1 - h) = 0.48 r_1^{1.3} h^{-0.3}$ и примем следующие обозначения: $z = \lg \Phi$; $x_1 = \lg a_z = \lg(100x_2)$; $a_z = \lg t$; $x_3 = \lg h$. После логарифмирования выражений и преобразований, получим общую линейную оптимизационную модель протяжки:

Максимизировать целевую функцию $z = x_1 - x_2 - (2 + \lg z_c) \rightarrow \max$
 при выполнении ограничений

$$y_1 = -\lambda x_1 + 0.79x_2 - (T_1 - \lg Q - \lg z_c) \geq 0;$$

$$y_2 = -\lambda x_1 + 0.79x_2 - (T_1 - \lg [\sigma_x] - \lg F_x - \lg z_c) \geq 0;$$

$$y_3 = -\lambda x_1 + 0.79x_2 - 0.6x_3 - (T_1 - \lg [\sigma_1] - 2.6 \lg r_1 + 0.1405 - \lg z_c) \geq 0;$$

$$y_4 = -x_1 + 2x_3 + (1.895 - \lg l_2 - \lg K_{\min}) \geq 0; y_5 = -x_2 + (\lg l_1 - \lg (z_{\min})) \geq 0;$$

$$y_6 = x_2 - (\lg l_2 - \lg (z_{\max})) \geq 0;$$

$$y_7 = x_2 - \lg t_{\min} \geq 0; y_8 = -x_3 - (0.7696 - \lg d) \geq 0;$$

$$y_9 = x_1 - \lg (100a_{z \min}) \geq 0; y_{10} = x_2 - x_3 - 0.3979 \geq 0;$$

где

$$T_1 = 0.2304 + \lg \pi + \lg d + 0.79 \lg l_2 + \lg C_p - 2\lambda + \lg K_p;$$

$K_p = K_\gamma K_c K_u$ – суммарный поправочный коэффициент силы резания

Полученная линейная модель позволяет определить для заданных размеров и материала детали подъем черновых зубьев a_z , шаг t , высоту h зубьев. Задача решается по известным правилам симплекс-метода.

Алгоритм определения принадлежности объекта к классу по методу ближайшего соседа

Сходство объектов лежит в основе алгоритма k-ближайших соседей [1]. Алгоритм способен выделить среди всех наблюдений k известных объектов (k-ближайших соседей), похожих на новый неизвестный ранее объект. На первом шаге алгоритма следует задать число k – количество ближайших соседей. На втором шаге находятся k записей с минимальным расстоянием до вектора признаков нового объекта (поиск соседей). Функция для расчета расстояния должна отвечать следующим правилам:

1. $d(x,y) \geq 0$, $d(x,y) = 0$ тогда и только тогда, когда $x = y$;
2. $d(x,y) = d(y,x)$;
3. $d(x,z) \leq d(x,y) + d(y,z)$, при условии, что точки x, y, z не лежат на одной прямой.

Для упорядоченных значений атрибутов находится Евклидово расстояние, где n – количество атрибутов:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Голоса за класс находятся по следующей формуле:

$$\text{Голос(класс)} = \sum_{i=1}^n \frac{1}{d^2(x, y_i)}$$

Класс, набравший наибольшее количество голосов, присуждается новой записи.

Решим задачу классификации для некоторых телефонных кабелей. Имеется набор данных о нескольких кабелях трех классов: ЗКП-1Ч4, КСПП-1Ч4, ПРППМ-1Ч2 [2]. Известны входные данные: диаметр жил и масса металла на 1 м длины, а выходное – класс телефонного кабеля.

Необходимо определить класс телефонного кабеля со следующими значениями диаметра жил и массы металла на 1м: 0,85 и 0,107. Зададим $k = 3$, ближайшими тремя соседями будут следующие телефонные кабели А(1,2; 0,26) – ЗКП-1Ч4, В(1,1; 0,22) – ЗКП-1Ч4, С(0,9; 0,106) – КСПП-1Ч4. Пусть имеется график зависимости диаметра жил разных кабелей от их массы, график имеет прямую зависимость. Расстояния между объектом и его ближайшими соседями равны:

$$d(\text{каб}; A) = \sqrt{(0,85 - 1,2)^2 + (0,107 - 0,26)^2} = \sqrt{0,1225 + 0,0234} = 0,38$$

$$d(\text{каб}; B) = \sqrt{(0,85 - 1,1)^2 + (0,107 - 0,22)^2} = \sqrt{0,0625 + 0,0128} = 0,27$$

$$d(\text{каб}; C) = \sqrt{(0,85 - 0,9)^2 + (0,107 - 0,106)^2} = \sqrt{0,0025 + 0,000001} = 0,05$$

Объединим полученные сведения в таблицу:

	Диаметр жил, мм	Масса металла на 1 м длины, кг	d	Класс
Ка б.	0,85	0,107	-	-
А	1,2	0,26	0,38	ЗКП-1Ч4
В	1,1	0,22	0,27	ЗКП-1Ч4
С	0,9	0,106	0,05	КСПП-1Ч4

Рассмотрим взвешенное голосование:

$$\text{Голос(ЗКП} - 1 \times 4) = \frac{1}{0,38^2} + \frac{1}{0,27^2} = \frac{1}{0,1444} + \frac{1}{0,0729} = 6,93 + 13,72 = 20,65$$

$$\text{Голос(КСПП} - 1 \times 4) = \frac{1}{0,05^2} = \frac{1}{0,0025} = 400$$

Так как $400 > 20,65$, то телефонный кабель классифицируется как КСПП-1Ч4

Литература

1. Алгоритм ближайшего соседа,
<http://www.basegroup.ru/library/analysis/regression/knn/>
2. Гроднев И.И. Линейные сооружения связи. М.:Радиосвязь.1987.

Оценивание параметров поля смещений рукописных символов

На современном этапе развития искусственные нейронные сети широко используются в задачах классификации образов и, в частности, в задачах классификации изображений, текстовой информации и машинного зрения [1, 2]. Решение задач классификации рукописного текста обуславливает возможность построения систем автоматического перевода его в стандартный символьный вид и, соответственно, возможность реализации систем распознавания ключевых слов в рукописном тексте и идентификации их автора. Решение этих задач, несомненно, сопряжено с необходимостью выбора типа нейронной сети, ее архитектуры, способов обучения и параметров сети.

Однако немаловажную роль играют также способы кодирования и предварительной обработки входных данных, обеспечивающие более эффективное функционирование нейронных сетей [3]. В основе указанной обработки могут лежать эвристические представления о характере входных данных и, в частности, о возможных видах их искажений. Алгоритмы, реализующие эту предварительную обработку, являются внешними по отношению к нейронной сети.

Предварительная обработка связана в первую очередь с устранением естественных отличий рукописных символов от их эталонных изображений. Эти отличия обуславливаются в первую очередь изменением масштаба символов, их смещением и поворотом, а в ряде случаев и деформацией, описываемой некоторым нелинейным отображением. Причиной нелинейной деформации, например, может быть изгиб сканируемой поверхности листа с текстом. Другой вид искажений обусловлен наличием шума, на фоне которого наблюдается текст. При этом целью настоящей работы является рассмотрение вопросов оценивания поля подобных искажений, обусловленных в первую очередь изменением масштаба, смещением и вращением символов.

Образом символа рукописного текста на плоскости может считаться множество точек или пикселей, заданных своими координатами. Каждый пиксель записывается при этом в виде пары чисел (x_l, y_l) . Тогда образ k – символа можно записать в виде

$$M_k = \{(x_{lk}, y_{lk}) \mid l = 1, 2, \dots, m_k\}.$$

Расстояние точки $d_r = (x_r, y_r)$ до символа M_k можно задать выражением

$$\text{dist}(d_r, M_k) = \min_{l=1, 2, \dots, m_k} \sqrt{(x_r - x_{lk})^2 + (y_r - y_{lk})^2}.$$

Расстояние между двумя образами описывается при этом выражением

$$\text{dist}(M_j, M_k) = \sum_{d_r \in M_j} \text{dist}(d_r, M_k).$$

В этом случае расстояние между эталонным образом, который будет обозначаться как M_k^* , и его искаженным изображением находится по формуле

$$\text{dist}(M_k, M_k^*) = \sum_{d_r \in M_k} \text{dist}(d_r, M_k^*).$$

Очевидно, что приведенные формулы являются чувствительными к искажениям, типа смещения и вращения. Искажения смещения, представляемые аффинным преобразованием координат, можно описать выражением

$$\text{dist}(M_k, M_k^*) = \sum_{d_r \in M_k^* + d_0} \text{dist}(d_r, M_k^*),$$

где $d_0 = (x_0, y_0)$ представляет вектор аффинного преобразования координат.

Аналогичным способ определяются искажения, обусловленные вращением символа. Пример такого вращения приведен на рис. 1. Вращение символа на угол φ задается выражением

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Если матрицу вращения обозначить символом $\mathbf{W}(\varphi)$, то искажения, обусловленные вращением, можно представить выражением

$$\text{dist}(M_k, M_k^*) = \sum_{d_r \in \mathbf{W}(\varphi)M_k^*} \text{dist}(d_r, M_k^*).$$

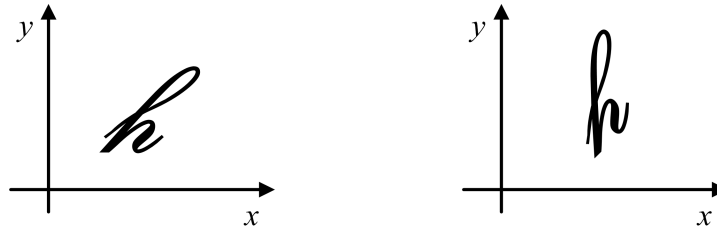


Рис. 1. Пример искажений, обусловленных вращением символа

Отсюда видно, что для обеспечения инвариантной оценки символов к их смещениям и вращениям в качестве расстояния между символами можно использовать выражения

$$\text{dist}(M_k, M_k^*) = \min_{d_0 \in D_k} \sum_{d_r \in M_k + d_0} \text{dist}(d_r, M_k^*) \text{ и}$$

$$\text{dist}(M_k, M_k^*) = \min_{\varphi \in \Omega_k} \sum_{d_r \in \mathbf{W}(\varphi)M_k} \text{dist}(d_r, M_k^*).$$

Здесь D_k и Ω_k обозначают диапазоны изменения параметров аффинного смещения и вращения, соответственно.

Оценивание образа по данным входного образа M_j осуществляется при этом посредством минимизации одновременно по смещению и вращению символа, что соответствует выражению

$$\mathcal{M}(M_j) = \min_k \min_{d_0 \in D_k, \varphi \in \Omega_k} \sum_{d_r \in \mathbf{W}(\varphi)(M_k^* + d_0)} \text{dist}(d_r, M_k^*).$$

Применение данного подхода предполагает, естественно, предварительную нормализацию входных данных по размеру образа. При этом приведенные выражения могут быть обобщены и на случай разномасштабного изменения размеров образа по координатным осям. Вполне очевидными являются и обобщения на случай нелинейной деформации изображений.

При практическом применении этих выражений могут быть применены различные упрощения, обеспечивающие эффективное вычисление оценок символов по наблюдаемым искаженным символам. Потребность в упрощениях диктуется в первую очередь необходимостью нахождения минимумов функций, определяющих расстояния между образами. Приведенные выражения решают задачу классификации, хотя в этой связи необходимы дополнительные исследования.

Представляется естественным, что данный подход применим не только при классификации символов рукописного текста, но и в более общих задачах анализа контурных изображений. В этом плане может найти применение и техника кодирования кривых комплексными числами, а в более общем случае, и техника кодирования в трехмерном пространстве с использованием кватернионов.

Литература

1. Хайкин С. Нейронные сети: полный курс. – М.: Издательский дом «Вильямс», 2006. – 1104 с.
2. 13th International Conference, CAIP 2009 Munster, Germany, September 2-4, 2009. Proceedings: Computer analysis of images and patterns. Editors Jiang X., Petkov N. – Berlin, Heidelberg: Springer, 2009, pp. 165 – 236.
3. Elden L. Matrix methods in data mining and pattern recognition. – Philadelphia: SIAM, 2007. – 224 p.

Численные методы решения дифференциальных уравнений

Методы точного решения дифференциальных уравнений позволяют найти решение только для ограниченного вида уравнений, большая часть уравнений, встречающихся на практике нельзя проинтегрировать с помощью этих методов.

В случаях, если решение невозможно найти в виде аналитической функции, применяются численные методы, которые представляют решение таких уравнений в виде таблиц функции, которую мы ищем в зависимости от переменной.

Есть несколько методов численного решения уравнений, отличающихся друг от друга по сложности и точности полученного результата.

Дано уравнение $y' = x + y$ при начальном условии $y(0) = 1$ на отрезке $[0; 0,5]$ с шагом 0,1. Вычисления сведены в таблицу для краткости и наглядности.

Метод Рунге-Кутты

i	x_i	k		Δy_i	y_i
0	0	1	0,1000	0,1104	1
		2	0,1100		
		3	0,1105		
		4	0,1211		
1	0,1	1	0,1210	0,1325	1,1104
		2	0,1321		
		3	0,1326		
		4	0,1443		
2	0,2	1	0,1443	0,1569	1,2429
		2	0,1565		
		3	0,1571		
		4	0,1700		
3	0,3	1	0,1700	0,1840	1,3998
		2	0,1835		
		3	0,1842		
		4	0,1984		
4	0,4	1	0,1984	0,2138	1,5838
		2	0,2133		
		3	0,2140		
		4	0,2298		
5	0,5				1,7976

Метод Эйлера.

i	0	1	2	3	4	5
x_i	0,0	0,1	0,2	0,3	0,4	0,5
y_i	1	1,1	1,22	1,362	1,528	1,721

Решения, найденные аналитическим методом.

Общее: $y = Ce^x - x - 1$

С учётом начального условия: $1 = C - 0 - 1; C = 2$

Частое решение: $y = 2e^x - x - 1$

Для сравнения полученные результаты сведены в таблицу.

i	x_i	y_i		
		Метод Эйлера	Метод Рунге-Кутты	Точное значение
0	0	1	1	1
1	0,1	1,1	1,1104	1,1103
2	0,2	1,22	1,2429	1,2428
3	0,3	1,362	1,3998	1,3997
4	0,4	1,528	1,5383	1,5837
5	0,5	1,721	1,7976	1,7975

Вывод:

Из таблицы очевидно, что метод Рунге – Кутты даёт наиболее точный ответ в сравнении с методом Эйлера (точность до 0,0001). Заметно, что погрешность возрастает с каждым шагом вычисления. Причиной этого является округление приближённого значения при каждом шаге и то, что в качестве начальных данных для каждого последующего вычисления используется предыдущий расчёт. Вследствие чего и накапливается погрешность.