Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет «Прикладная математика и физика»

Кафедра «Вычислительная математика и информатика»

Курсовой проект по

Информатике и вычислительной технике по теме: «Процедуры и функции в качестве параметров»

Выполнил: Щербаков А.А.

Студент группы М8О-106Б

Преподаватель: Дубинин А.В.

Оценка:

Дата:

Москва 2016

Введение

Человечество постоянно развивается и задачи, возникающие перед учеными, становятся все сложнее и сложнее. Так с развитием математики появилась надобность в решении не алгебраических, а трансцендентных уравнений. Для таких уравнений практически невозможно установить точное значение корня, но чаще всего для практических целей нам такое излишество и ни к чему. Пять, шесть знаков после запятой - этого бывает более, чем достаточно, когда речь идёт о проектировке сложного технического устройства (а обычно и того меньше). Ручной подсчет корней трансцендентных уравнений займет невероятное количество времени, поэтому ставится задача проведения данных расчетов при помощи компьютеров. Например, зная диапазон, в котором находится корень, можно найти его приближенное значение. В данной работе будут рассмотрены 3 основных метода, разработанных с этой целью:

1. Метод дихотомии (половинного деления)
2. Метод итераций
3. Метод Ньютона (частный случай метода итераций)

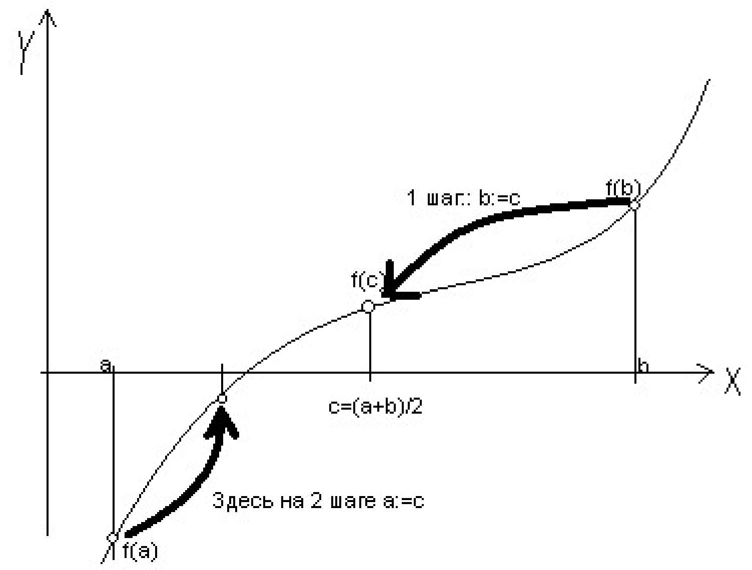
Метод дихотомии

Пусть задан отрезок [a, b], на котором необходимо найти корень какого – либо уравнения. Так как функция монотонна на данном отрезке и на нём есть корень, то значения функции в точках a и b имеют разные знаки (F(a)*F(b)<0). Далее вычисления проводятся по формулам:

1. Если F($a^{(k)}$)*F$\left(\frac{a^{(k)}+b^{(k)}}{2}\right)$>0, то $a^{(k+1)} = \frac{a^{(k)}+b^{(k)}}{2}$, $b^{(k+1)} = b^{(k)}$.
2. Если F($b^{(k)}$)*F$\left(\frac{a^{(k)}+b^{(k)}}{2}\right)$>0, то $a^{(k+1)} = a^{(k)}$, $b^{(k+1)} = \frac{a^{(k)}+b^{(k)}}{2}$.

Данный процесс повторяется до тех пор, пока не будет выполнено условие окончания $\left|a^k - b^k\right| < Eps$, где Eps – достаточно малая величина. Таким образом, приближенное значение корня:

$$X \approx \frac{a^{(\text{конечное})}+b^{(\text{конечное})}}{2}$$



Метод итераций

Идея метода заключается в замене исходного уравнения F(x)=0 уравнением вида x=f(x).

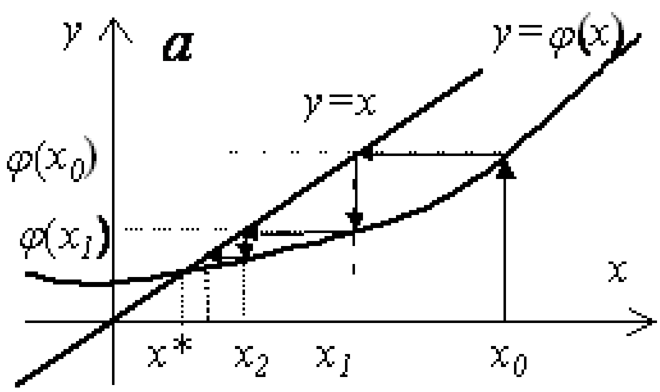
Достаточное условие сходимости метода: $|f'(x)|<1$, $x \in [a, b]$. Если это условие не выполняется, то метод расходится и им нельзя найти приближенное значение корня на данном отрезке.

Начальное приближение корня: $x^{(0)} = \frac{a+b}{2}$

Итерационный процесс: $x^{(k+1)} = f\left(x^{(k)}\right)$.

Условие окончания: $|x^{(k)} - x^{(k-1)}|<Eps$.

Приближенное значение корня $x \approx x^{(\text{конечное})}$.



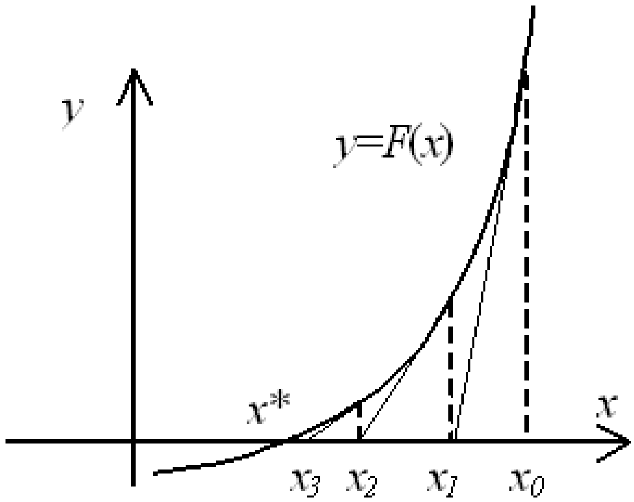
Метод Ньютона

Данный метод является частным случаем метода итераций.

Условие сходимости метода: $|F'(x) * F''(x)| < \left(F'(x)\right)^2$.

Итерационный процесс: $x^{(k+1)} = x^{(k)} - \frac{F\left(x^{(k)}\right)}{F'\left(x^{(k)}\right)}$.

Как и в предыдущем методе, при невыполнении условия сходимости метод расходится.



Вычисление производных:

Для вычисления значения производной в точке используется формула: $f'(x) = \frac{f(x+h)-f(x-h)}{2h}$, где h – достаточно малая величина.

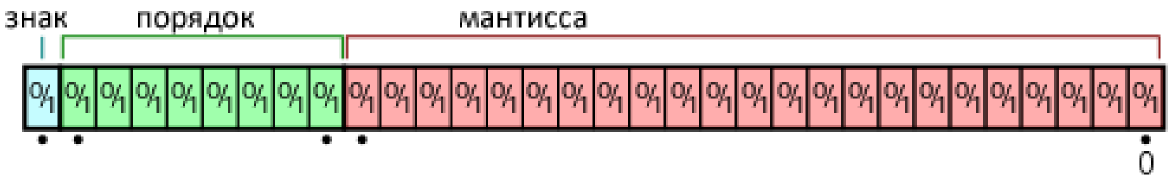
Значение второй производной в точке будет вычисляться следующим образом: $f'\left(f'(x)\right) = \frac{f(x+h)+f(x-h)-2f(x)}{h*h}$

Если выбрать значение Eps и h слишком малыми, то возможно бесконечное зацикливание в некоторых методах. Поэтому необходимо выбирать их относительно большими так, чтобы значение корня было максимально точно.

Числа с плавающей точкой

Множество целых чисел бесконечно, но мы всегда можем подобрать такое число бит, чтобы представить любое целое число, возникающее при решении конкретной задачи. Множество действительных чисел не только бесконечно, но еще и непрерывно, поэтому, сколько бы мы не взяли бит, мы неизбежно столкнемся с числами, которые не имеют точного представления. Числа с плавающей запятой — один из возможных способов представления действительных чисел, который является компромиссом между точностью и диапазоном принимаемых значений.

Число с плавающей запятой состоит из набора отдельных разрядов, условно разделенных на знак, порядок и мантиссу. Порядок и мантисса — целые числа, которые вместе со знаком дают представление числа с плавающей запятой в следующем виде:



Математически это записывается так:

$(-1)^s \times M \times B^E$, где s - знак, B=2 - основание, E -порядок, а M -мантисса.

Так как в все вычисления производятся в двоичной системе счисления, то B = 2. Для экономии места и достижения однозначности представления чисел было решено представлять числа в нормализованном виде и не хранить в мантиссе старшую единицу.

Строго говоря, нормализованное число имеет следующий вид:
$(-1)^s \times 1.M \times 2^E$.

Особенности арифметики

1. Округление:
    - Округление до ближайшего в стандарте сделано не так как мы привыкли. Математически показано, что если 0,5 округлять до 1 (в большую сторону), то существует набор операций, при которых ошибка округления будет возрастать до бесконечности. Поэтому в IEEE754 применяется правило округления до четного. Так, 12,5 будет округлено до 12, а 13,5 – до 14.

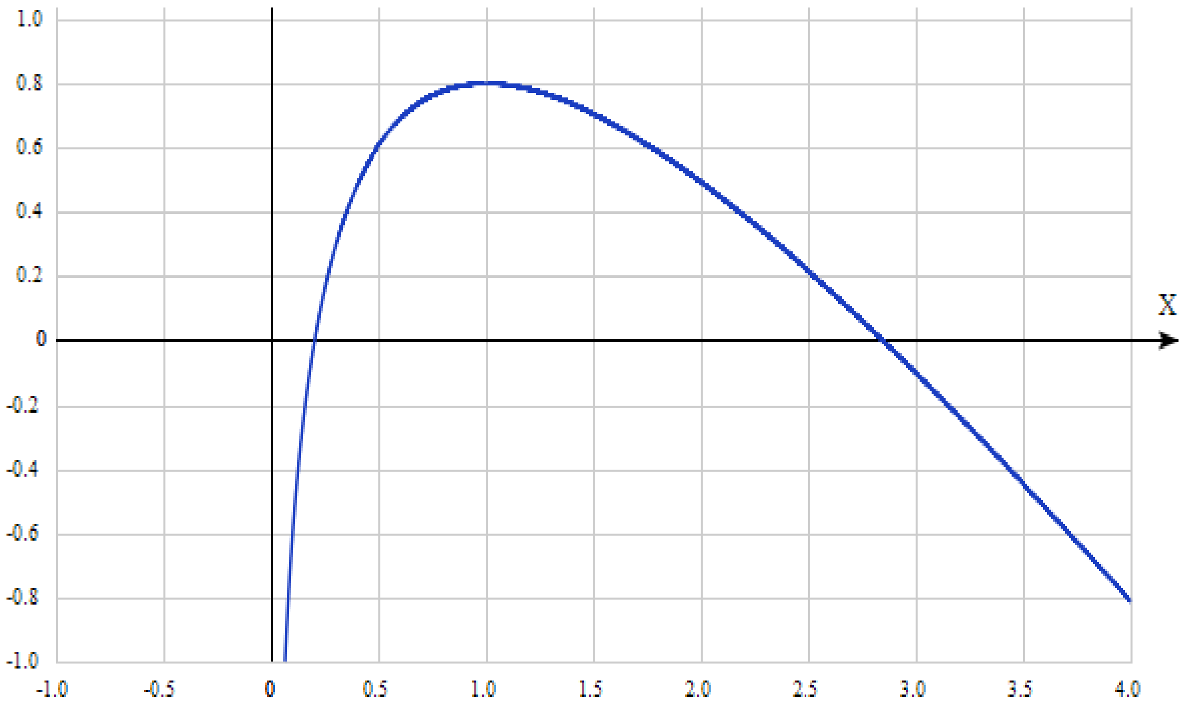
- Округление до ближайшего в стандарте сделано не так как мы привыкли. Математически показано, что если 0,5 округлять до 1 (в большую сторону), то существует набор операций, при которых ошибка округления будет возрастать до бесконечности. Поэтому в IEEE754 применяется правило округления до четного. Так, 12,5 будет округлено до 12, а 13,5 – до 14.

2. Неассоциативность арифметических операций
3. Не все десятичные числа имеют двоичное представление с плавающей запятой. Например, число «0,2» будет представлено как «0,200000003» в одинарной точности
4. Из-за особенностей представления чисел их нельзя сравнивать “ в лоб”.

Выполнение работы

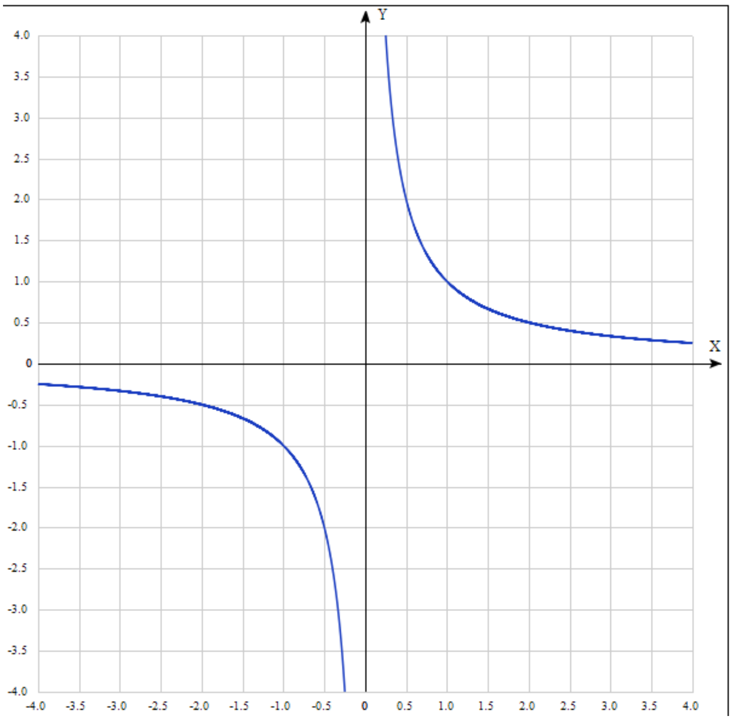
Вариант 12: **<u>F(x) = ln(x) – x + 1,8</u>** на отрезке **<u>[2;3]</u>**

Построим исходный график F(x):



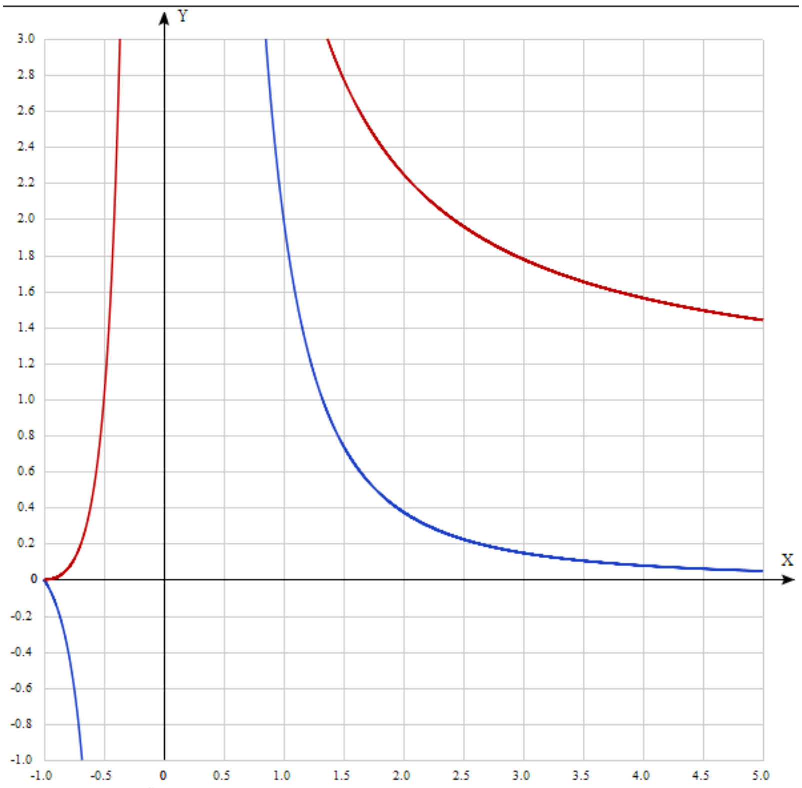
Преобразуем функцию к виду x = ф(x).

График ф’(x) = (ln(x)+1,8)’ = 1/x:



На отрезке [2;3] график проходит ниже прямой y=1, т.е. условие сходимости метода итераций выполняется.

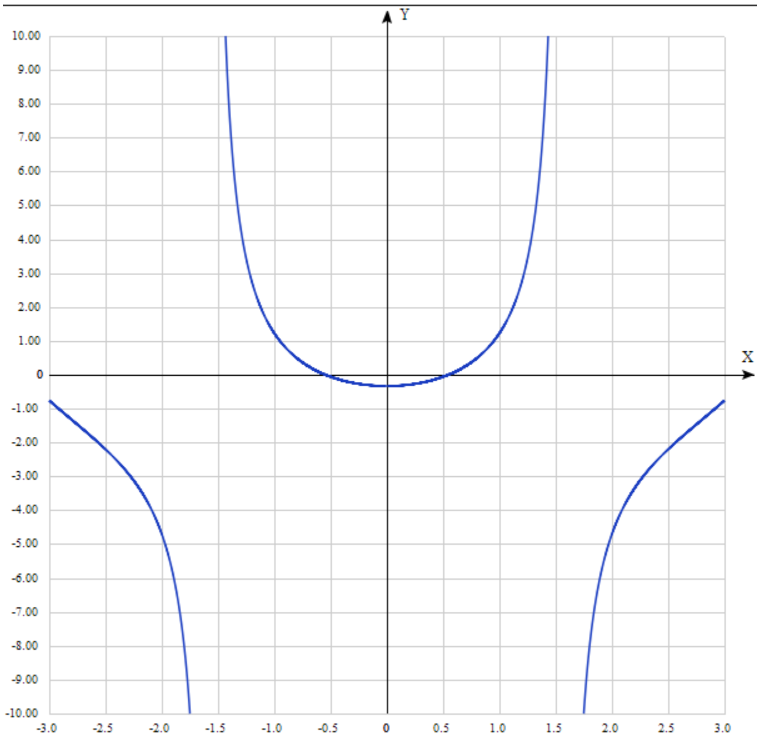
Построим графики | F’(x) * F”(x) | и (F’(x))^2:



Приходим к выводу, что метод Ньютона на отрезке [2;3] сходится.

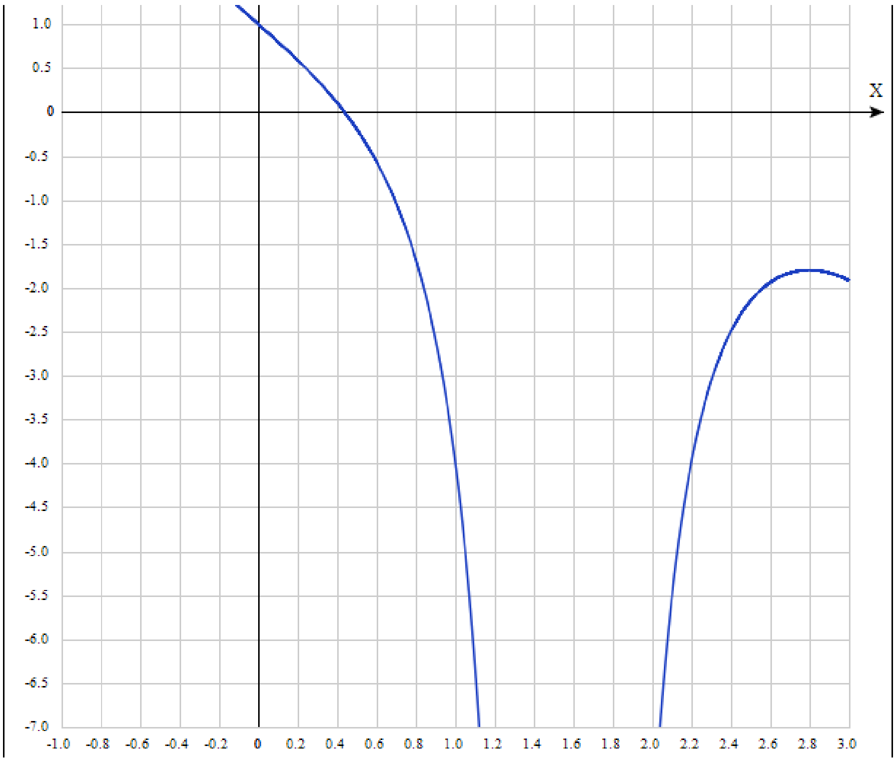
Вариант 13: **F(x) = x*tg(x)-1/3** на отрезке **[0.2;1]**

Построим исходный график F(x):



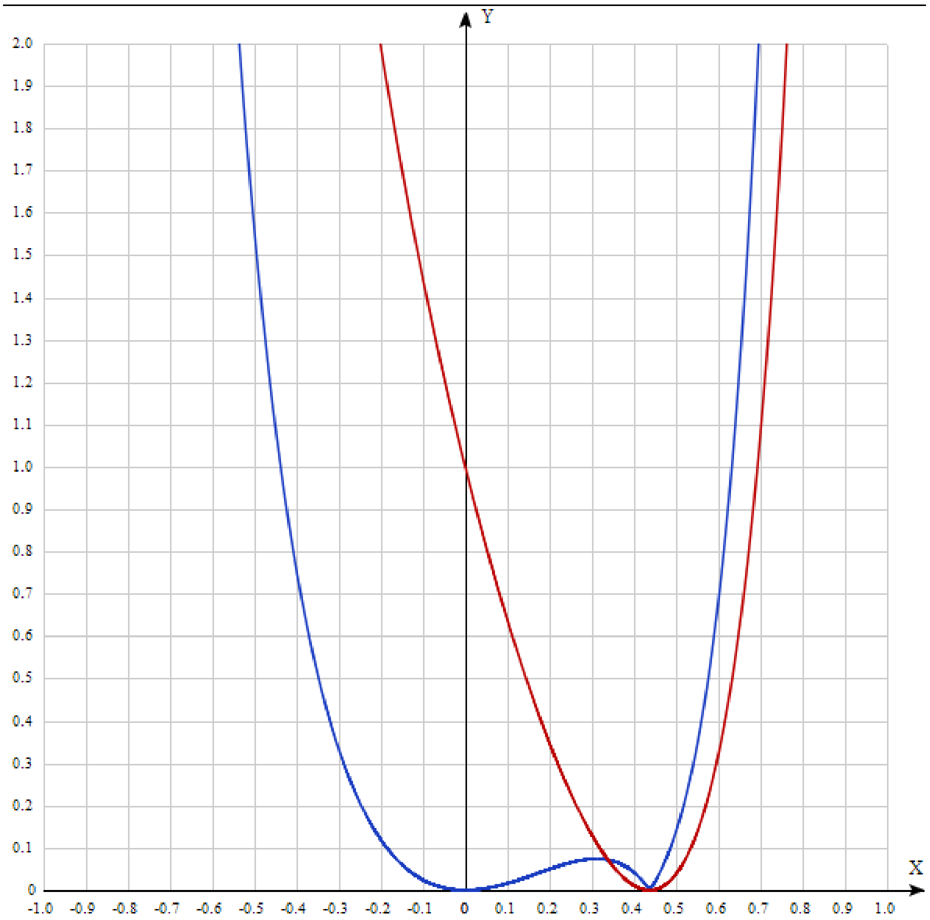
Преобразуем функцию к виду x = ф(x).

График ф'(x) = (x- x*tg(x)-1/3)' = 1-tg(x)-x*$\cos^{-2}$(x):



На отрезке [0.2;1] график проходит ниже прямой y=1, т.е. условие сходимости метода итераций выполняется.

Построим графики | F’(x) * F”(x) | и (F’(x))^ 2:



Условие сходимости не выполняется на всём отрезке, но метод не расходится в некоторой области искомого значения, значит для такой функции метод Ньютона может быть применим.

Вывод

Можно прийти к выводу, что описанные методы действенны не для каждой функции. Самый нетребовательный из них – метод дихотомии, который не имеет условий сходимости, поэтому может быть применен для более обширного круга задач, нежели методы итераций и Ньютона, которые накладывают достаточно жесткие ограничения на исследуемую функцию. Но, как бы то ни было, все описанные методы работают, и вполне могут быть применены на практике, ведь с их помощью можно получить результат быстрее, чем используя многие другие численные методы.

Также, при работе с действительными числами в памяти компьютера, не стоит забывать об особенностях чисел с плавающей точкой. Сейчас арифметика с плавающей точкой почти совершенна: практически всегда наивный подход сработает, и программа, не учитывающая все ее особенности, выдаст правильный результат. Но нужно всегда оставаться бдительным: в таком вопросе как компьютерная математика легко наступить на грабли.