**Лабораторная работа 1**

**Вычисления с плавающей точкой**

[**Результаты**](https://refnew.ru/?q=%D0%A0%D0%B5%D0%B7%D1%83%D0%BB%D1%8C%D1%82%D0%B0%D1%82%D1%8B)**вычислений:**https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_11852162.gif  
https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_m46c833b7.png  
Также, просчитав вручную первые три члена обоих алгоритмов, убедимся, что они были запрограммированы правильно.

Для алгоритма с вычислением факториала:

https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_m4b2d7c19.gif  
https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_m6bf74fd4.gif  
https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_4382421a.gif

Для алгоритма без вычисления факториала:

https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_m4b2d7c19.gif  
https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_m17e69697.gif  
https://refnew.ru/laboratornaya-rabota-1-vichisleniya-s-plavayushej-tochkoj-furs/3563_html_35229eaf.gif  
**Выводы:**при вычислении суммы в предоставленной по условию форме происходит быстрый рост слагаемых ряда, что приводит к выходу значений за границы допустимого диапазона. Следовательно, необходимо было произвести ряд преобразований, чтобы избавиться от n!, т.к. факториал становится причиной быстрого роста значения, что в свою очередь влечет за собой переполнения.

Проведём анализ полученных результатов:

Была ли катастрофическая потеря верных знаков? Чтобы это проверить, сравним максимальные по модулю члены рядов суммы каждого алгоритма с результатами, полученными в результате работы каждой программы. Для вычислений без факториала:

189.80035714285710

0.9999779095029978

Здесь мы видим, что доверять можно лишь первым 13 верным знакам потому, что максимальное значение хранит 17 верных знаков (произошла катастрофическая потеря верных знаков) и со сто процентной вероятностью можно сказать, что в 14 знаке происходило округления (и, возможно, в других знаках).

В алгоритме с вычислением факториала получаем следующее:

189.80035714285714

0.9999779095030239

Здесь ситуация аналогична предыдущему алгоритму.

В итоге можно заключить, что алгоритм с вычислением факториала производит вычисления на порядок точнее. Однако, если для области, в которой применяется результат erf(x) функции, достаточно 11 верных знаков (учитывая округления и накопление ошибок в результате), то подойдут оба алгоритма.