π Project

Math 242, spring 2020

due Friday, February 14

Implement *at least two* of the following methods for computing digits of π. For each implementation, you should write a Mathematica module that accepts one parameter, the number of terms or iterations to compute. Your module should return an approximation of π.

- Ramanujan (1910):
$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801}\sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26\,390k)}{(k!)^4(396)^{4k}}$$
- Chudnovsky brothers (1989):
$$\frac{1}{\pi} = 12\sum_{k=0}^{\infty} \frac{(-1)^k(6k)!(13\,591\,409 + 545\,140\,134k)}{(3k)!(k!)^3(640\,320)^{3k+3/2}}$$
- Salamin and Brent (1976): Set $a_0 = 1$, $b_0 = \frac{1}{\sqrt{2}}$, $s_0 = \frac{1}{2}$. Then iterate:
$$a_k = \frac{a_{k-1} + b_{k-1}}{2}, \quad b_k = \sqrt{a_{k-1}b_{k-1}}, \quad c_k = a_k^2 - b_k^2, \quad s_k = s_{k-1} - 2^k c_k, \quad p_k = \frac{2a_k^2}{s_k}$$
Then p_k gives an approximation to π.

For each method that you implement, make a plot that compares the number of correct digits produced by each algorithm, for various numbers of terms or iterations. Look for patterns and determine how many correct digits of π are produced by each term of the summation or each iteration of the formulas.

Turn in a Mathematica notebook that contains not only your code, but also explanation of what your code does. You don't have to explain where the formulas come from, but make it clear that you know how your implementations work. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course). Submit your notebook on the Moodle page for this course.

Your notebook will be graded on a scale of 0 to 16 points. The following rubric gives characteristics of notebooks that will merit sample point totals. (Interpolate the following for point totals that are not divisible by 4.)

16 points. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and conjectures are discussed. Notebook is well-formatted and easy to read.

12 points. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and conjectures.

8 points. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

4 points. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

0 points. Notebook is not turned in.