Introduction to Languages for Scientific Computing

Prof. **Paolo Bientinesi**

pauldj@aices.rwth-aachen.de

High Performance and Automatic Computing

RWTH AACHEN UNIVERSITY

Challenge: bit-flips

Back in the 60's, the computers on the starship Enterprise used what is known as the 10-bit "Star Trek" arithmetic with normalization:

$$\beta = 2, \quad t = 7, \quad e_{\min} = -3, \quad e_{\max} = 4;$$

this arithmetic does not include subnormal numbers, NaNs, infinities, underflows, overflows and such[1].

During the many intergalactic journeys, the Enterprise was exposed to all sorts of cosmic radiations, some of which were known to cause the unfortunate "bit-flip" phenomenon: every so often, one (and only one) bit of a number would flip from zero to one or vice-versa. As one can expect, bit-flips were the sources of unexpected and unwanted system behaviours. As an example, because of the radiations the number represented by $[0101010, 101]$ could become $[0111010, 101]$.

The objective of this challenge is to measure the average error caused by the bit-flips for the specific case of the number pi.

[1] The three-digit exponent defines the integers $[0, \ldots, 7]$, which logically map to $[-3, \ldots, 4]$.

Bit-flips 2/2

- **Step 1:** Let π_{ST} be the “Star Trek” representation of π.
- **Step 2:** Consider all the possible ways that bit-flips can affect π_{ST} and compute Σ, the average of the relative errors caused by them.
- **Step 3:** Return the 10 digits of Σ_{ST}, corresponding to the representation of Σ in the Star Trek arithmetic.

Rules

The first student who sends me the right answer wins the challenge.

- Individual assignment
- Submit both the final answer and its derivation
- Submission by email to `pauldj@aices.rwth-aachen.de`
- Email's subject: "`LSC-15 Challenge1 <your last name>`"
- Accepted formats: plain text, pdf
- Name your file `<your name>.txt` or `<your name>.pdf`
- **Deadline: The challenge is open until solved**