Advanced practical **Programming** for Scientists

Thorsten Koch

Zuse Institute Berlin

TU Berlin

SS2017

The Zen of Python, by Tim Peters (Part 2)

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

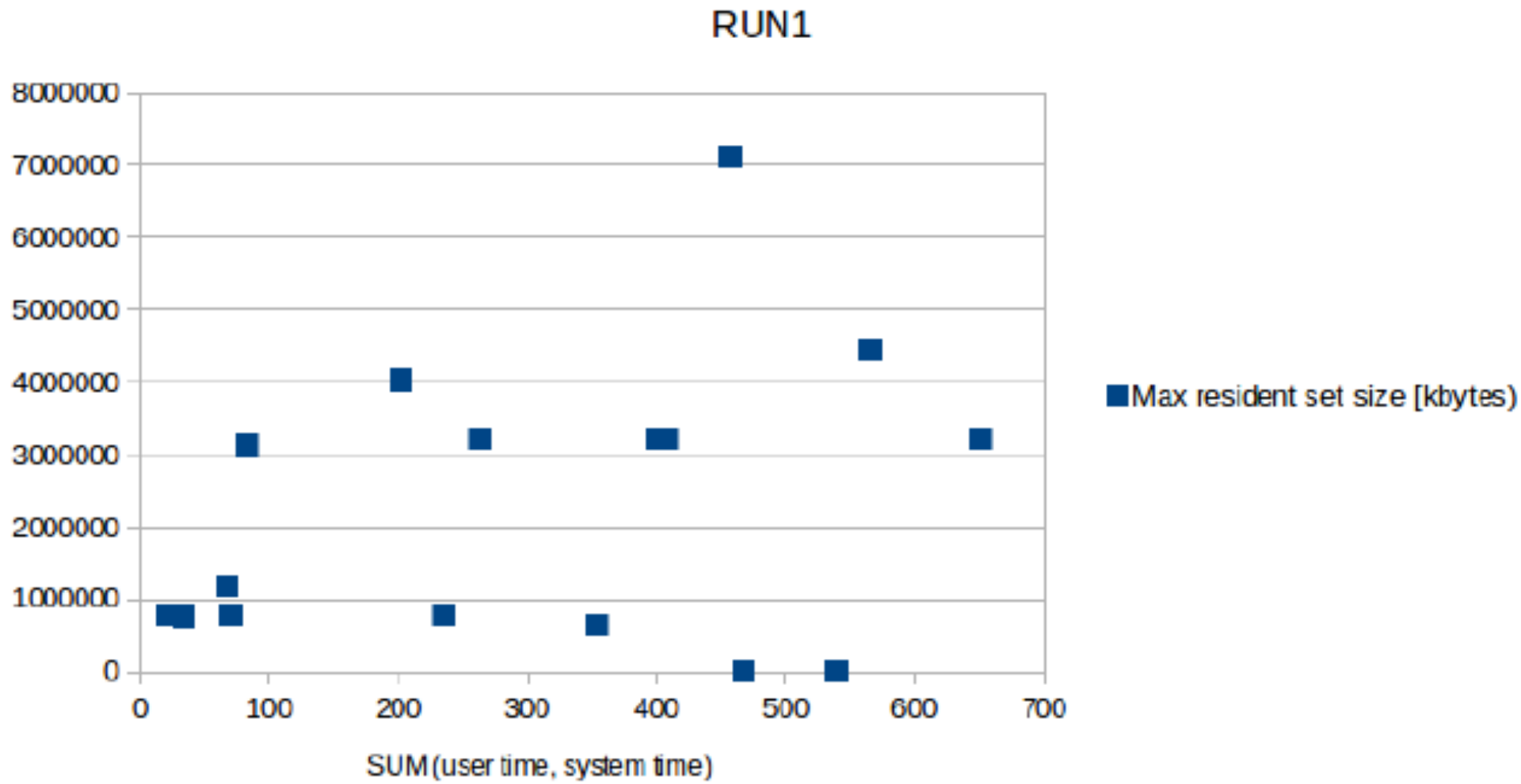
Although never is often better than *right* now.

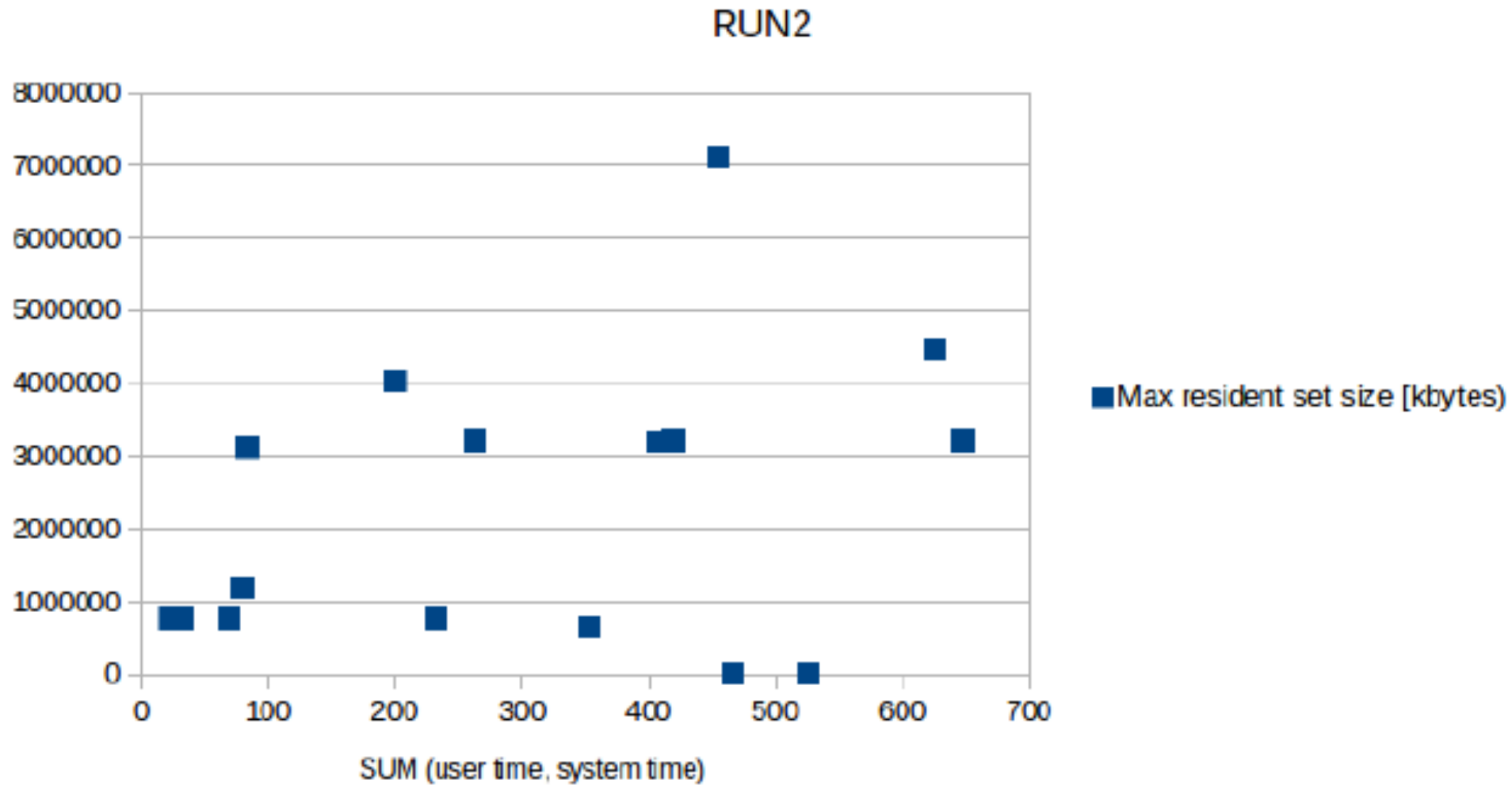
If the implementation is hard to explain, it's a bad idea.

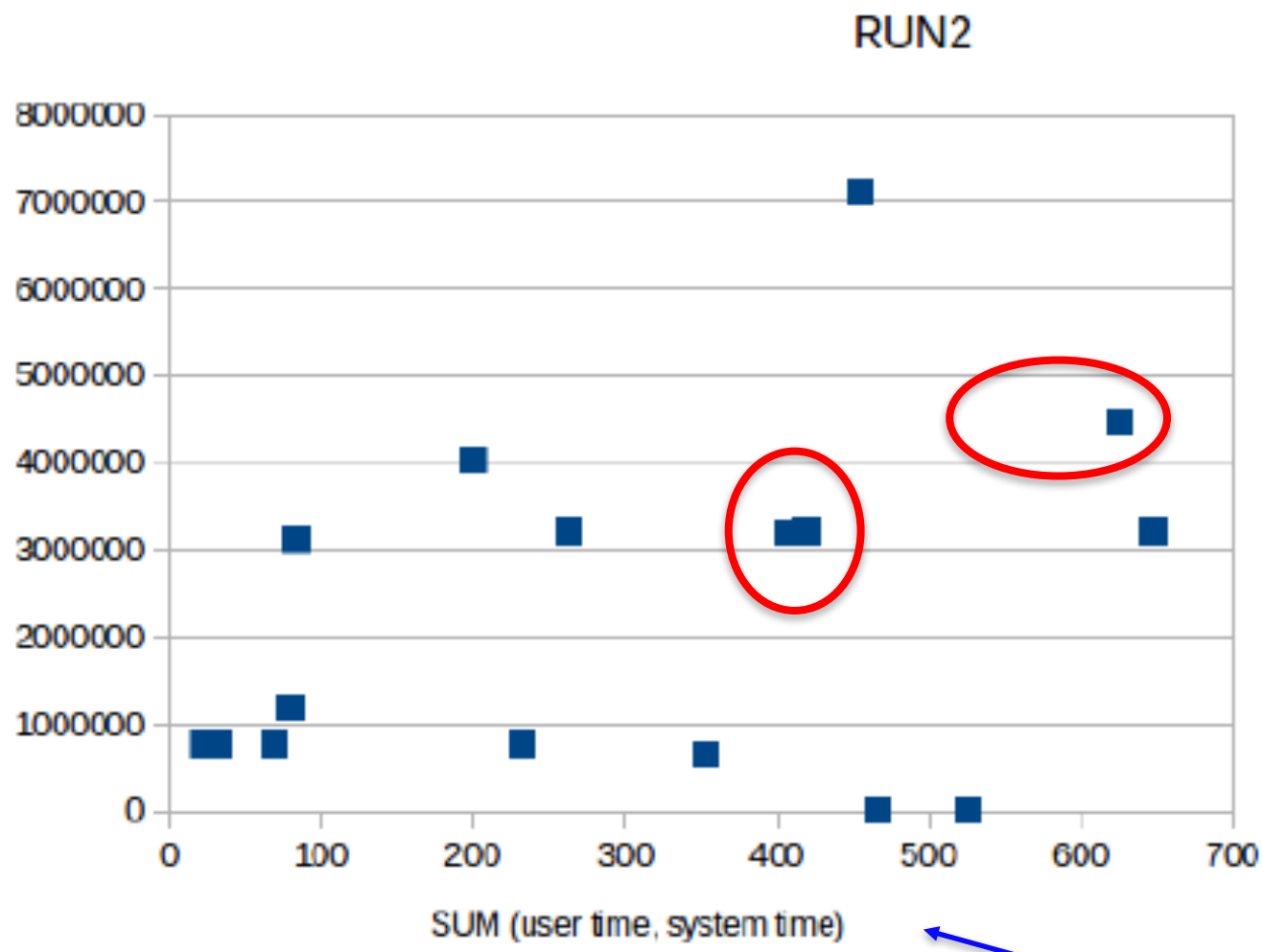
If the implementation is easy to explain, it may be a good idea.

User		User Time	System Time	Summe	Elapsed Time	Differenz	Max resident set size [kbytes]
ex01.cpp	CPP	21,07	0,61	21,68	21,69	-0,01	790896
ex1.c	C#	32,44	0,7	33,14	33,15	-0,01	786532
ex1.py	PY	399,5	1,72	401,22	401,28	-0,06	3222252
AmeyVasulkar							
Baumann	CPP	233,96	0,57	234,53	234,58	-0,05	792836
Brand	CPP	69,55	0,76	70,31	70,33	-0,02	792728
christopher_wyczisk	PY	261,86	1,58	263,44	263,44	0	3225776
Dinsel	PY	199,49	1,92	201,41	201,44	-0,03	4038312
Fleischer							
hark	PY	537,74	0,74	538,48	538,64	-0,16	9044
Jeney	PY	648,8	1,76	650,56	650,62	-0,06	3235052
josefluispelz	PY	405,62	1,8	407,42	407,47	-0,05	3225268
Julia							
Jung							
marthakarpeter	PY	453,11	3,81	456,92	459,5	-2,58	7114184
mikulski_patryk							
MoritzMorgenroth							
RayChew	CPP	66,57	0,96	67,53	67,54	-0,01	1201164
rettkowski							
richter							
Sanny							
schloesser	PY	352,56	1,81	354,37	354,37	0	653674
Setje-Eilers	PY	563,12	2,46	565,58	565,75	-0,17	4454388
Shrive	C#	81,32	1,53	82,85	82,85	0	3127120
Sterling							
Tam							
tapia	PY	466,39	0,59	466,98	466,96	0,02	7472
Viernickel							
Wegscheider	CPP	67,64	0,61	68,25	68,26	-0,01	791480

User Time	System Time	Summe	Elapsed Time	Differenz	Max resident set size [kbytes]
21,12	0,5	21,62	21,68	-0,06	792968
31,34	0,98	32,32	32,32	0	784436
417,8	1,92	419,72	419,79	-0,07	3222404
231,59	0,56	232,15	232,19	-0,04	792776
68,47	0,67	69,14	69,14	0	791764
261,29	1,48	262,77	262,9	-0,13	3224440
197,73	1,89	199,62	199,63	-0,01	4038348
523,85	0,56	524,41	524,42	-0,01	9020
645,86	1,68	647,54	647,72	-0,18	3225252
404,09	1,72	405,81	405,82	-0,01	3215460
451,87	3,59	455,46	455,59	-0,13	7114104
79,01	0,82	79,83	79,89	-0,06	1204108
351,03	1,79	352,82	353,08	-0,26	653596
623,19	2,44	625,63	625,79	-0,16	4454300
81,39	1,52	82,91	82,93	-0,02	3127112
465,79	0,52	466,31	466,7	-0,39	7440
67,71	0,66	68,37	68,38	-0,01	792800


RUN1
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
0
100
200
300
400
500
600
700
SUM (user time, system time)
Max resident set size [kbytes]


RUN2
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
0
100
200
300
400
500
600
700
SUM (user time, system time)
Max resident set size [kbytes)



GB

Max resident set size [kbytes]

]

seconds

User		User Time	System Time	Summe	Elapsed Time	Differenz	Max resident set size [kbytes]
ex01.cpp	CPP	21,07	0,61	21,68	21,69	-0,01	790896
ex1.c	C#	32,44	0,7	33,14	33,15	-0,01	786532
ex1.py	PY	399,5	1,72	401,22	401,28	-0,06	3222252
AmeyVasulkar							
Baumann	CPP	233,96	0,57	234,53	234,58	-0,05	792836
Brand	CPP	69,55	0,76	70,31	70,33	-0,02	792728
christopher_wyczisk	PY	261,86	1,58	263,44	263,44	0	3225776
Dinsel	PY	199,49	1,92	201,41	201,44	-0,03	4038312
Fleischer							
hark	PY	537,74	0,74	538,48	538,64	-0,16	9044
Jeney	PY	648,8	1,76	650,56	650,62	-0,06	3235052
josefluispelz	PY	405,62	1,8	407,42	407,47	-0,05	3225268
Julia							
Jung							
marthakarpeter	PY	453,11	3,81	456,92	459,5	-2,58	7114184
mikulski_patryk							
MoritzMorgenroth							
RayChew	CPP	66,57	0,96	67,53	67,54	-0,01	1201164
rettkowski							
richter							
Sanny							
schloesser	PY	352,56	1,81	354,37	354,37	0	653674
Setje-Eilers	PY	563,12	2,46	565,58	565,75	-0,17	4454388
Shrive	C#	81,32	1,53	82,85	82,85	0	3127120
Sterling							
Tam							
tapia	PY	466,39	0,59	466,98	466,96	0,02	7472
Viernickel							
Wegscheider	CPP	67,64	0,61	68,25	68,26	-0,01	791480

Experiment

An **experiment** is a procedure carried out to support, refute, or validate a hypothesis. Experiments provide insight into cause-and-effect by demonstrating what outcome occurs when a particular factor is manipulated. Experiments vary greatly in goal and scale, but always rely on repeatable procedure and logical analysis of the results.

An experiment usually tests a hypothesis, which is an expectation about how a particular process or phenomenon works. However, an experiment may also aim to answer a "what-if" question, without a specific expectation about what the experiment reveals, or to confirm prior results. If an experiment is carefully conducted, the results usually either support or disprove the hypothesis. […] An experiment must also control the possible confounding factors—any factors that would mar the accuracy or repeatability of the experiment or the ability to interpret the results. Confounding is commonly eliminated through scientific controls and/or, in randomized experiments, through random assignment.

https://en.wikipedia.org/w/index.php?title=Experiment&oldid=781092819

Automatize!

- You have to computational results again, anyway.
- Try to automatize them as much as possible.
- Check whether the results of successive runs fit.
- Sample if neccessary

resident set size (**RSS**) is the portion of memory occupied by a process that is held in main memory (RAM). The rest of the occupied memory exists in the swap space or file system, either because some parts of the occupied memory were paged out, or because some parts of the executable were never loaded.[

https://en.wikipedia.org/w/index.php?title=Resident_set_size&oldid=767184487

1;1;17976931348623157081452742373170435679807056752584499659891747680315726078002853876058955863276687817154045895351438246423432132688946418276846754670353751698604991057655128207624549009038932894407586850845513394230458323690322294816580855933212334827479782620414472316873817718091929988125040402618412485836 8.000000

2;1;1.797693e+308

3;1;0.000000

4;1;2147483647

5;1;9223372036854775807

6;1;0.0000000000000001

7;1;1e-307

8;1;1e-308

9;1;0.9e-307

10;1;0.3e-307

11;1;0.2e-307

```
Line 3: Invalid value
Line 8: Invalid value
Line 11: Invalid value
File: test4.dat with 11 lines
Valid values Loc0: 8 with GeoMean: 0.000000
Valid values Loc1: 0 with GeoMean: 1.000000
```

1;1;17976931348623157081452742373170435679807056752584499
65989174768031572607800285387605895586327668781715404589
53514382464234321326889464182768467546703537516986049910
57655128207624549009038932894407586850845513394230458323
69032229481658085593321233482747978262041447231687381771
8091929988125040402618412485836 9.000000

```
File: test5.dat with 1 lines
Valid values Loc0: 1 with GeoMean:
17976931348623157081452742373170435679807056752584499659
89174768031572607800285387605895586327668781715404589535
14382464234321326889464182768467546703537516986049910576
55128207624549009038932894407586850845513394230458323690
32229481658085593321233482747978262041447231687381771809
19299881250404026184124858368.000000
Valid values Loc1: 0 with GeoMean: 1.000000


File: test5.dat with 1 lines
ex1b: ex1b.c:68: geom_mean_log: Assertion
`!fetestexcept(FE_INVALID | FE_DIVBYZERO | FE_OVERFLOW |
FE_UNDERFLOW)' failed.
Aborted (core dumped)
```

```c
static double geom_mean_log(const double* const x, const size_t count)
{
   assert(NULL != x);

   double sum = 0.0;

   // We do this as an asswert, because we do not expect an error to be possible
   assert(!feclearexcept(FE_ALL_EXCEPT));

   for(size_t i = 0; i < count; i++) // note: due to unsigned  while(count >= 0) does not work
   {
      assert(!isnan(x[i]) && !islessequal(x[i], 0.0));

      sum += log2(x[i]);
   }
   double gm = exp2(sum / (double)count);

   // We do this as an asswert, because we do not expect an error to be possible. Or?
   assert(!fetestexcept(FE_INVALID | FE_DIVBYZERO | FE_OVERFLOW | FE_UNDERFLOW));

   return gm;
}
```

Errors should never pass silently

(unless explicitly silenced)

Ariane 5

On 4 June 1996 the maiden flight of the Ariane 5 launcher ended in a failure, about 40 seconds after initiation of the flight sequence. At an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded. The failure was caused by "*complete loss of guidance and attitude information*" 30 seconds after liftoff.

The problem was caused by an `Operand Error' in converting data in a subroutine from 64-bit floating point to 16-bit signed integer. One value was too large to be converted, creating the Operand Error. This was not explicitly handled in the program (although other potential Operand Errors were) and so the computer, the Inertial Reference System (SRI) halted, as specified in other requirements. There are two SRIs, one `active', one `hot back-up' and the active one halted just after the backup, from the same problem. Since no inertial guidance was now available, and the control system depends on it, we can say that the destructive consequence was the result of `Garbage in, garbage out' (GIGO). The conversion error occurred in a routine which had been reused from the Ariane 4 vehicle, whose launch trajectory was different from that of the Ariane 5. The variable containing the calculation of Horizontal Bias (BH), a quantity related to the horizontal velocity, thus went out of `planned' bounds (`planned' for the Ariane 4) and caused the Operand Error.

Requirements and Testing

a) The operand range in the module was deliberately not protected;

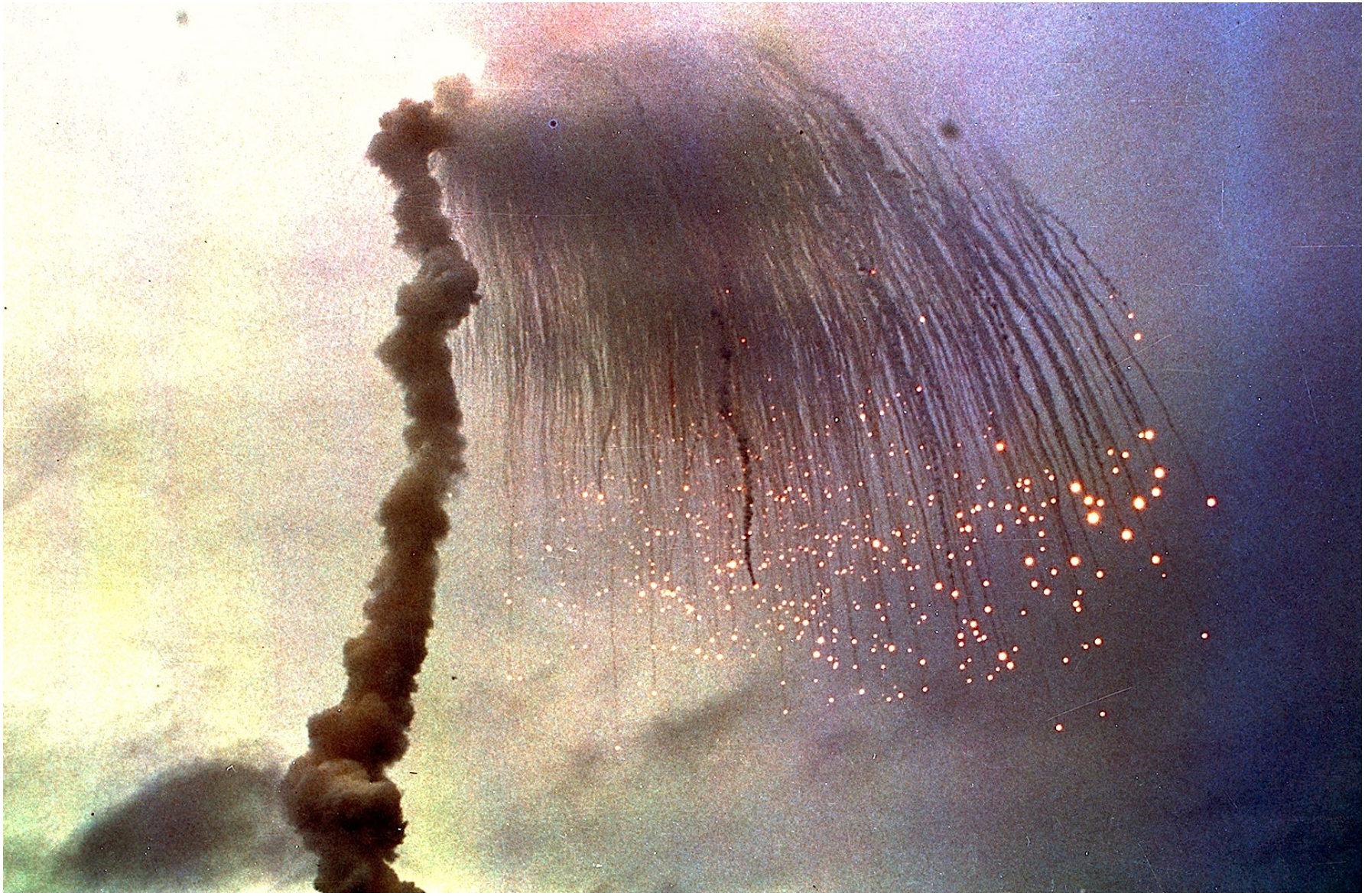
b) this was because engineering analysis for its use in Ariane 4 had shown the operand would never go out of bounds;

c) the range requirement stemming from this analysis was not transferred to the requirements for the Ariane 5;

d) testing was done against requirements

this is more properly classified as a requirements error rather than a programming error. The program was written against Ariane 4 requirements; these requirements were not transferred to the Ariane 5 requirements spec; the Ariane 5 requirements therefore did not state the range requirement; the (implicit in Ariane 5) range requirement was in conflict with the behavior of Ariane 5 (as in fact explicated in other Ariane 5 requirements); requirements came up against behavior and the rocket was destroyed. (It is not surprising that it was a requirements error - over 90% of safety-critical systems failures are requirements errors, according to a JPL study that has become folklore)

Ariane 5



Static Software Analysis

C/C++

http://cppcheck.sourceforge.net/

https://clang.llvm.org/docs/index.html

http://valgrind.org/

http://www.gimpel.com/html/flex.htm (commercial)

https://www.grammatech.com/products/codesonar (commercial)

http://ltp.sourceforge.net/coverage/lcov.php

Python

https://coverage.readthedocs.io/en/coverage-4.4.1/

https://www.pylint.org/

https://blog.codacy.com/review-of-python-static-analysis-tools-ff8e7e27f972

C/C++, Java, Python, etc.

https://scan.coverity.com/ (commercial, free use)

Ada

http://www.adacore.com/codepeer

Exercise

Read http://matt.might.net/articles/intro-to-make/ and have a look at http://berrendorf.inf.h-brs.de/sonstiges/make.html

1. Ex1:
   a. check for consecutive sequence numbers
   b. check boundary values
2. Ex1,Ex2:
   a. document using doxygen or similar
   b. write makefile to generate documentation with: `make doc`
   c. Write makefile to generate coverage with: `make coverage`
   d. Write makefile to run static checks: `make check`
3. `Ex3:`
   We will write a program to predict a time series
   you will find data on Monday on github.
   Write a program that reads in the data and predicts the next 24 values.