Numerical Analysis of Source Codes in Thales Entities

Romain Soulat – Thales Research & Technology
Floating points are hard
So what are floating point numbers?

- Roughly base 2 scientific notation
  
  + 0.6022141 \cdot 10^{24}
  
  ↓
  
  0 24 6022141

- More formally
  
  \[ f = s \cdot 2^e \cdot m \]

- Stored into 32 bits with
  
  - 1 bit for sign
  - 8 bits for exponent (able to store -126 to 127)
  - 23 bits for mantissa (able to store ~6 significant digits in base 10)
  - Special values of exponent are used for ±\infty, NaN
Issues: approximation of floats in machine

```c
#include <stdio.h>

int main(){
    float res = 0;

    for (int i=0; i < 1000; i++){
        res = res + 0.1;
    }

    printf("1000*0.1 = %f\n", res);
    return 0;
}
```

1000*0.1 = 99.999046
What can go wrong?

```
#include <stdio.h>

int main(){
    float res = 0;
    for (int i=0;i<1000; i++){
        res = res + 0.1;
    }
    printf("%f", res);
    return 0;
}
```

99.999046

Slow loss of precision

```
#include <stdio.h>

int main(){
    float t1 = 1000000.1f;
    float t2 = 1000000.2f;
    float pos1 = 900.0f;
    float pos2 = 910.0f;
    float speed =
        fabs(pos2-pos1)/(t2 - t1);
    printf("%f\n", speed);
    return 0;
}
```

160.00000

Quick loss of precision

```
#include <stdio.h>

int main(){
    float a = 0.5f;
    a = a - 0.3f;
    a = a - 0.2f;
    if (a >= 0.0f){
        printf("%.10f",a);
    }
    else{
        printf("%.10f",100.0f*a+13.37);
    }
}
```

13.369998509

Unstable tests
Thales markets

- Aerospace
- Space
- Ground Transport
- Defense
- Cybersecurity

Thales Markets

Aerospace
Space
Ground Transport
Defense
Cybersecurity

Thales Research & Technology France
Template trp 8.0.2 / 87211168-GRP-EN-003
But there’s hope!
Large body work on the subject!

13 tools tested during first year of work

➢ From academia, PMEs
➢ Open source benchmarks, Thales entities code
➢ Objective criteria: level of guarantee, scalability, ...
➢ « Thales » criteria: easy to use, insert in test suites, ...
General Toolchain

**Code**

```
int main(){
...
return 0;
}
```

**Tests**

```
... Float t=0; ...
```

**Makefile**

```
SOURCES = ...;
g++ -o ...
```

**Numeric precision requirements**

**Instrumented code**

```
DECLARE_RESOURCES
int main(){
...
END_MAIN
return 0;
}
```

**Instrumented Tests**

```
float t = FBETWEEN(-5,5);
... Float t=0;
```

**Ensure**

**Numeric stability**

**General verification**

(\textit{may use several tools})

**Open Source**

Verrou, FLDLib, Cadna, …

**Code and tests general instrumentation**

**Requires**

**Makesfile**

```
SOURCES = ...;
g++ -o $(OBJ_ANALYSIS)
```
What we provide to Thales developers

Analysis as a Service

VM with toolchain

All open source tools installed