Numerische Reproduzierbarkeit und parallele Berechnungen

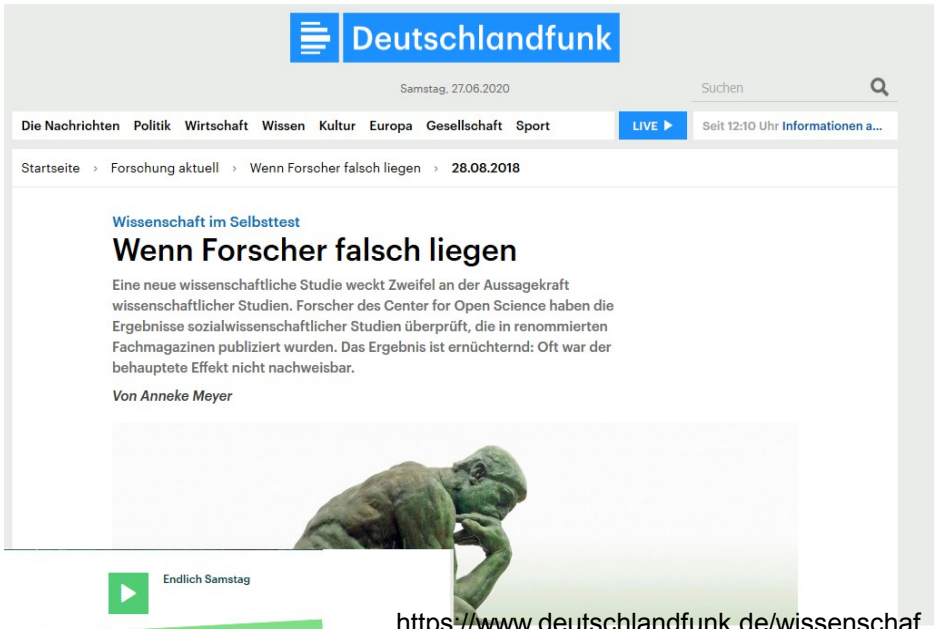
Softwareentwicklung in der Wissenschaft

Karl Ihlenfeldt

Gliederung

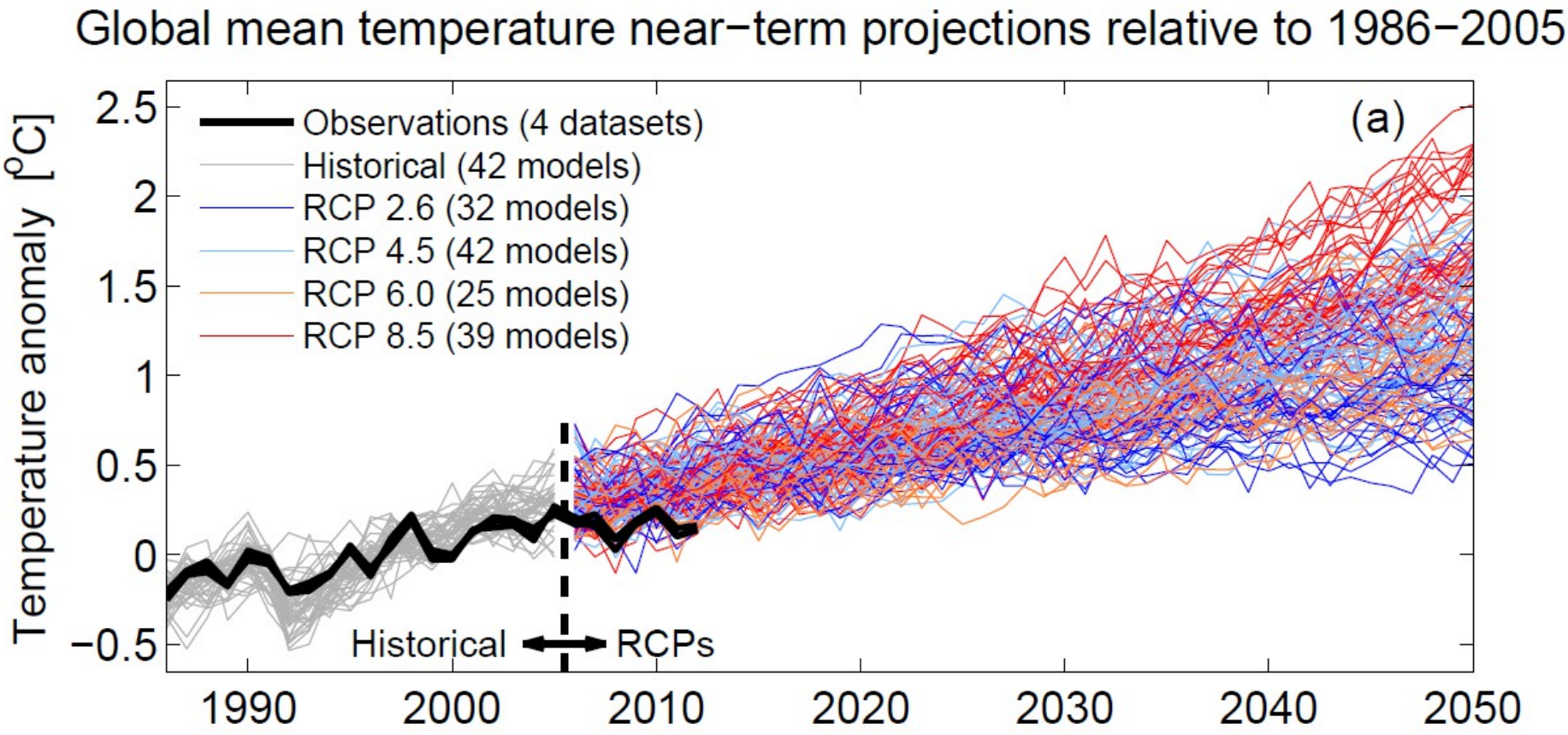
- Einleitung/Kontext
- Begriffserklärungen
- Gleitkommazahlen
  - Assoziativität
  - Rundungen
  - FMA
- Zwischenfazit
- Intervalle
  - Arithmetik
  - Abhängigkeit
  - Genauigkeit
- Fazit
- Quellen

https://www.wiko-greifswald.de/programm/vortraege/aktuelle-schwerpunkte/vertraue-keiner-statistik-glaubwuerdigkeit-der-wissenschaft/

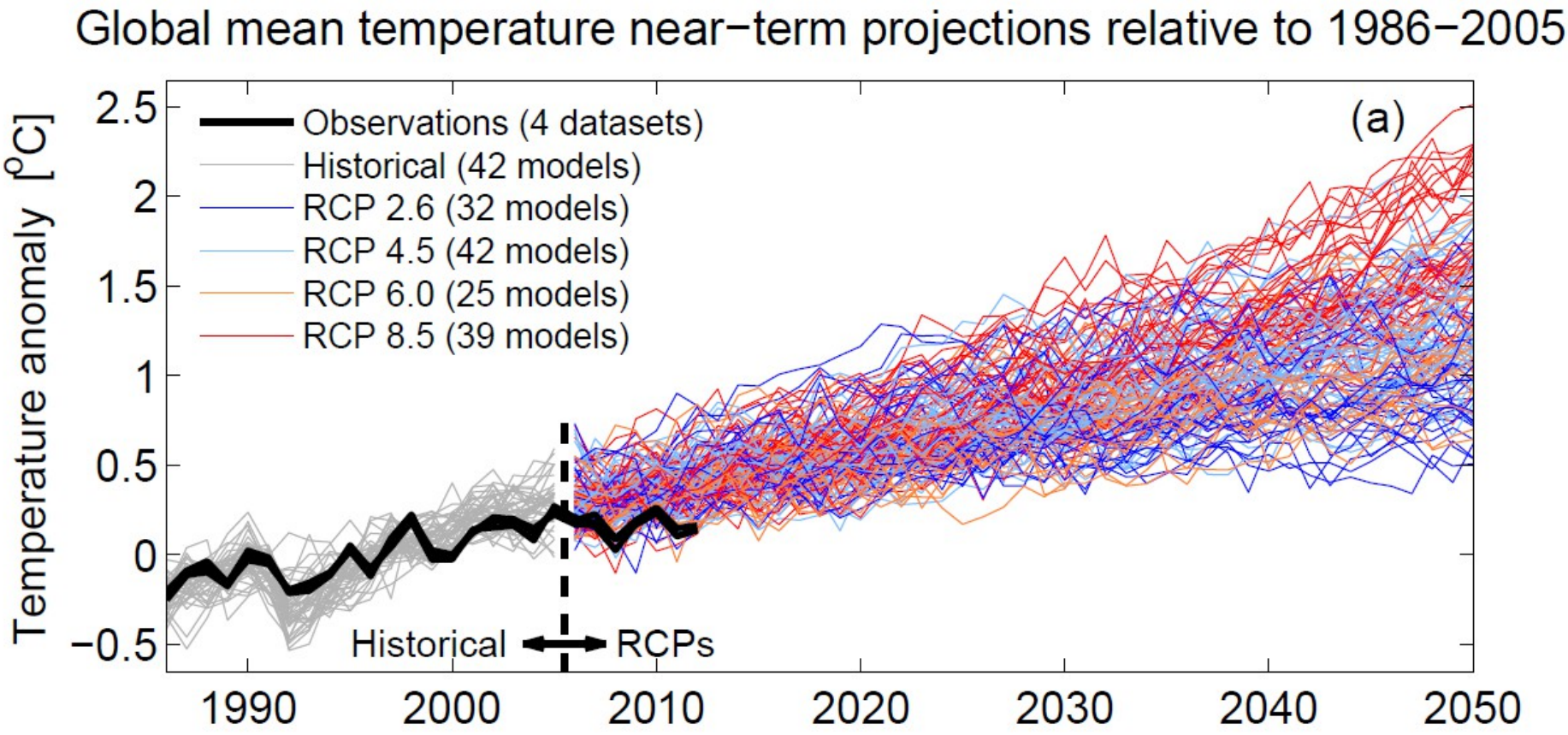


https://www.deutschlandfunk.de/wissenschaft-im-selbsttest-wenn-forscher-falsch-liegen.676.de.html?dram:article_id=426652

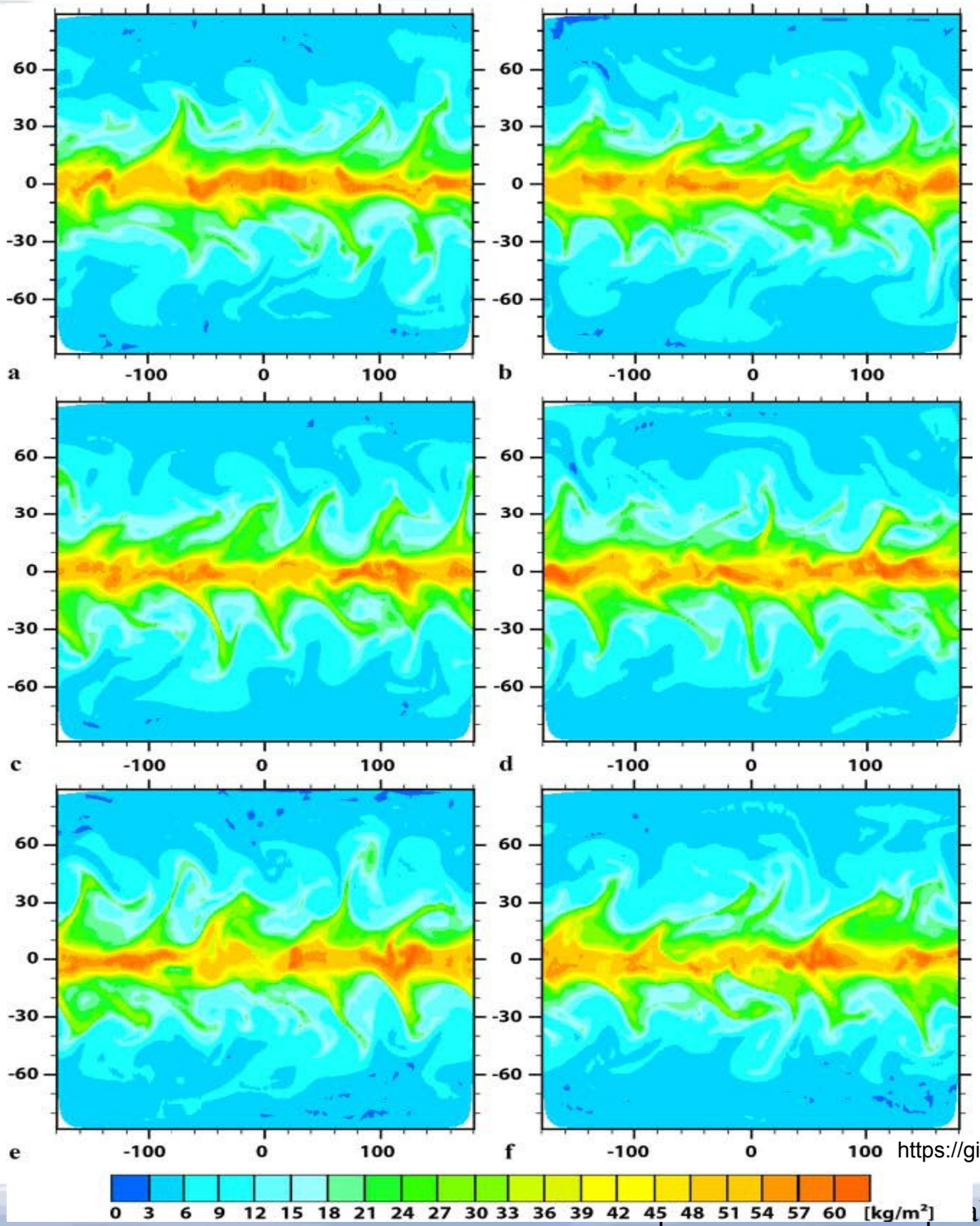
https://www.deutschlandfunknova.de/beitrag/reproduzierbarkeit-gleiche-studie-anderes-ergebnis



https://fee-misc.s3.amazonaws.com/files/imglib/20150424_ar5fig1125a.png



https://fee-misc.s3.amazonaws.com/files/imglib/20150424_ar5fig1125a.png



https://gi.de/informatiklexikon/reproduzierbarkeit

HPC

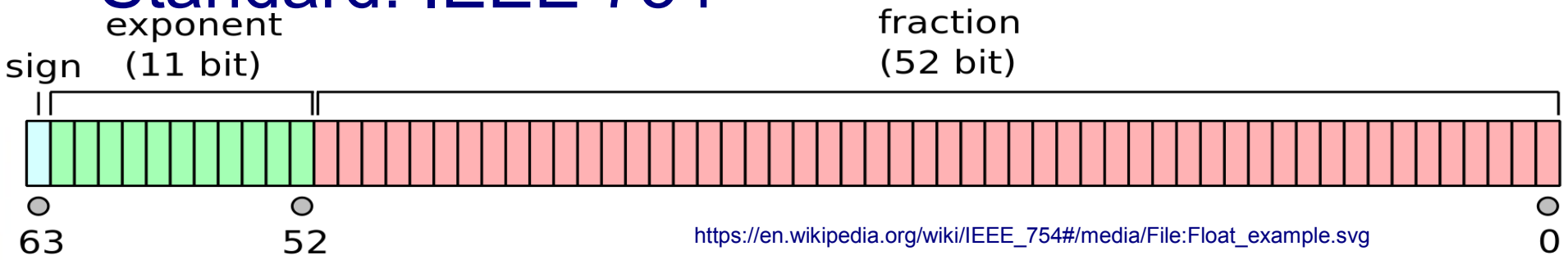
- Zusammenführung verschiedener Hardwarearchitekturen
- Parallele Berechnungen
  - Effizienzsteigerung
  - Mögliche Probleme:
    - Deadlock
    - Race-Condition
    - Nicht deterministische Reihenfolge

Numerische Reproduzierbarkeit

- Gleitkommazahlen
- Reproduzierbarkeit von Genauigkeit der Berechnung abhängig
  - Niedrige Genauigkeit ist weniger Aussagekräftig
  - Geschwindigkeit leidet unter hoher Genauigkeit
- Korrektheit des Ergebnisses schwer verifizierbar

Gleitkommazahlen

- Standard: IEEE 754



https://en.wikipedia.org/wiki/IEEE_754#/media/File:Float_example.svg

- Nicht assoziativ!

(fehlende) Assoziativität

$(1 + 2^{100}) - 2^{100} = 0$

$1 + (2^{100} - 2^{100}) = 1$

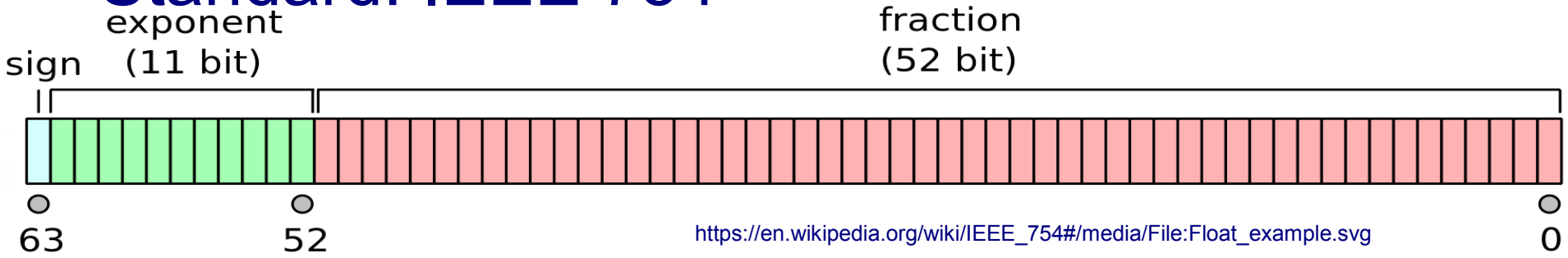
- $(1 + 2^{100})$ katastrophale Absorption
- $(1 + 2^{100}) - 2^{100}$ katastrophale Auslöschung

- Reihenfolge der Berechnungen nicht deterministisch

  → sporadisch auftretende Fehler
- Höhere Präzision für Fehlererkennung

Gleitkommazahlen

- Standard: IEEE 754



- Nicht assoziativ!
- Nicht alle Zahlen sind darstellbar
  - 0,2 = 0,00110011001100...
- Standardisierte Rundungsmodi

Standardisierte Rundungsmodi: IEEE 754

- In Richtung +∞
- In Richtung -∞
- In Richtung 0

Standardisierte Rundungsmodi: IEEE 754

- Zur nächsten darstellbaren Zahl, bei Unentschiedenheit:
  - Weg von 0
  - Zu gerade
    - Niedrigstes Bit wird 0
    - Default für die meisten Sprachen

Exkursion: BigDecimal

- Gleitkommazahl wird zu ganzer Zahl skaliert
- double a → BigDecimal($a*10^{scale}$)
- Addition/Subtraktion: Zahlen werden verrechnet, größere Skalierung übernommen
- Multiplikation/Division: Zahlen und Skalierung werden verrechnet
- Rundungsmodus wählbar
- 2 zusätzliche nicht-standardisierte Modi

Main.java

```java
import java.math.BigDecimal;
import java.math.RoundingMode;

public class Main {

    public static void main(String[] args) {

        BigDecimal a = new BigDecimal(1.2d);
        BigDecimal b = new BigDecimal(1.2d);
        System.out.println("a = b = " +a);

        a = a.setScale(5, RoundingMode.HALF_EVEN);
        System.out.println("Rundungsmodus: zu gerade: " +a);

        b = b.setScale(2, RoundingMode.DOWN);
        System.out.println("Rundungsmodus: zu 0: " +b);

        BigDecimal c = a.add(b);
        System.out.println("a + b = " +c);

        c = a.subtract(b);
        System.out.println("a - b = " +c);

        c = a.multiply(b);
        System.out.println("a * b = " +c);

        c = a.divide(b, RoundingMode.HALF_EVEN);
        System.out.println("a / b = " +c);
    }

}
```

Console | Problems | Debug Shell

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe
a = b = 1.19999999999999995559107901499373838305473327636718 75
Rundungsmodus: zu gerade: 1.20000
Rundungsmodus: zu 0: 1.19
a + b = 2.39000
a - b = 0.01000
a * b = 1.4280000
a / b = 1.00840
```

Bibliotheken für Rundungsmodi

- C und C++
  - fenv.h
- Fortran
  - libbfp754
- Ruby/Java
  - BigDecimal
- Python
  - Gmpy2

FMA

- Fused Multiply-Addition
- Result = a*b + c

  Ohne FMA:

  - result = (a*b), result wird gerundet
  - result += c wird addiert, result wird gerundet

  Mit FMA:

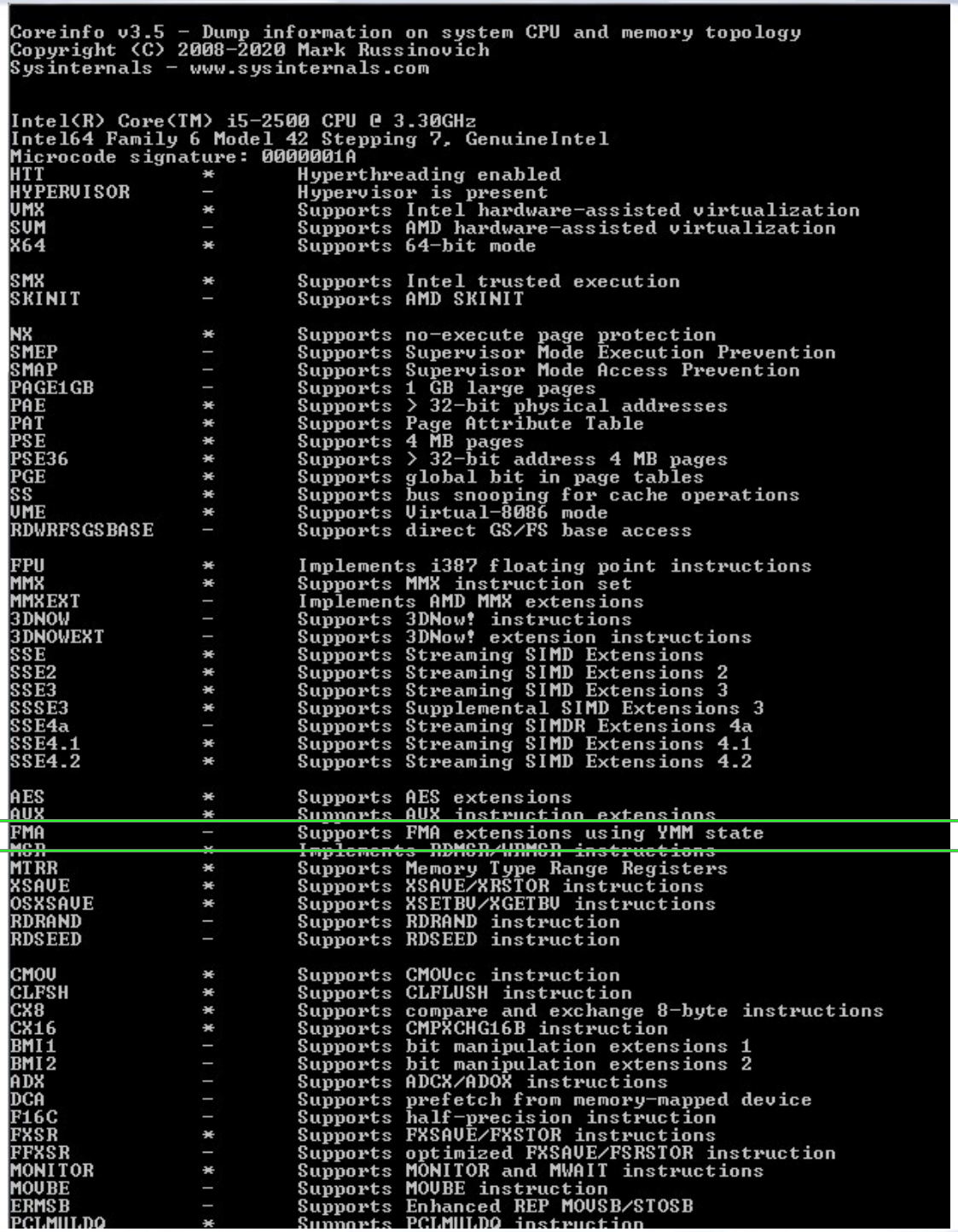
  - result = (a*b)+c, result wird gerundet

FMA

- Instruction Set für Prozessoren
  - Erweiterung für 128 und 256-bit SIMD
- Methode in vielen math-Bibliotheken

FMA

- Abhängig von Soft- und Hardware
- Implementiert von Intel und AMD
  - Coreinfo
- Compiler/Assembler müssen FMA unterstützen
  - GNU CC
  - Intel Compilers
  - ...

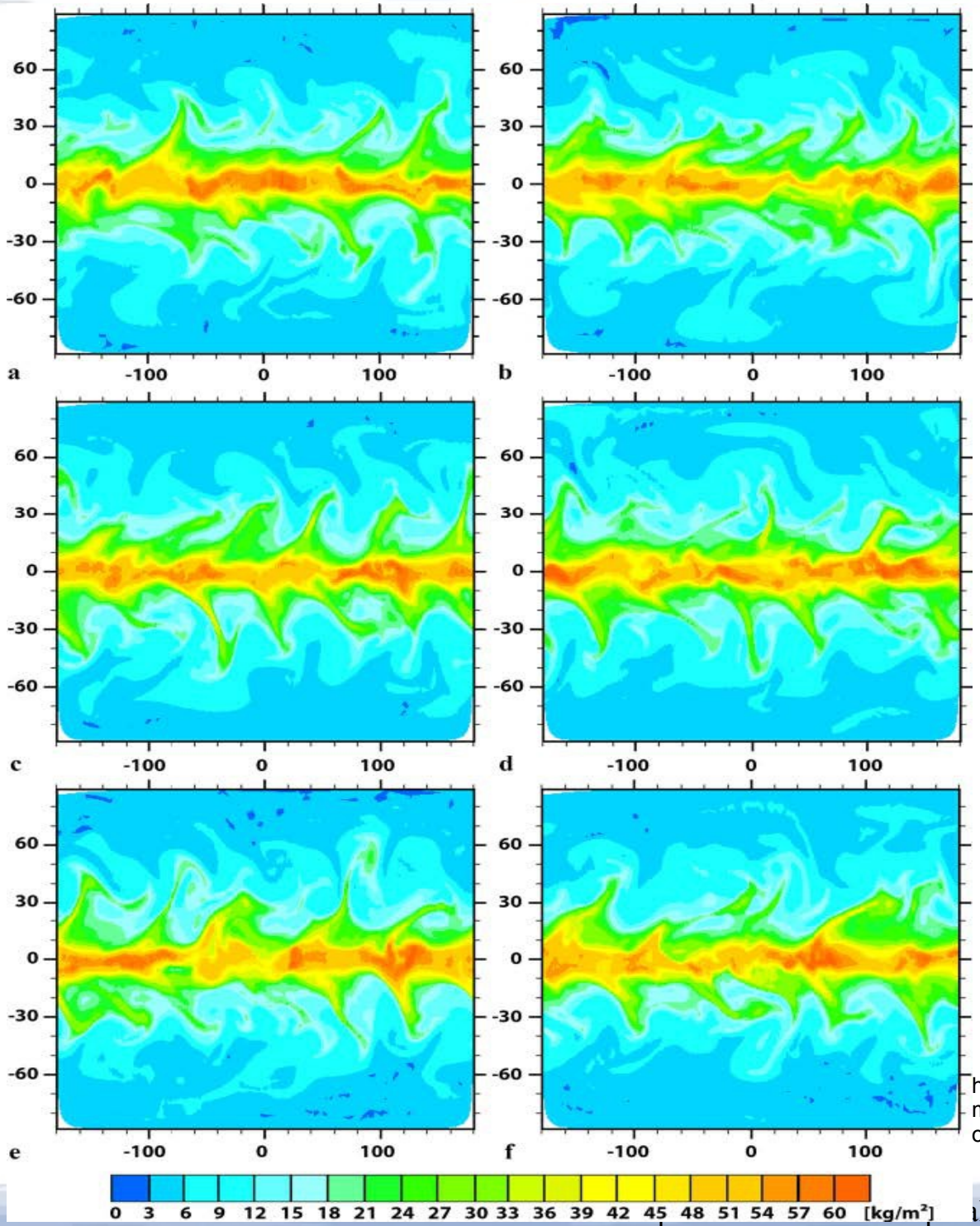

Coreinfo v3.5 - Dump information on system CPU and memory topology
Copyright (C) 2008-2020 Mark Russinovich
Sysinternals - www.sysinternals.com
Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz
Intel64 Family 6 Model 42 Stepping 7, GenuineIntel
Microcode signature: 0000001A
HTT * Hyperthreading enabled
HYPERVISOR - Hypervisor is present
VMX * Supports Intel hardware-assisted virtualization
SVM - Supports AMD hardware-assisted virtualization
X64 * Supports 64-bit mode
SMX * Supports Intel trusted execution
SKINIT - Supports AMD SKINIT
NX * Supports no-execute page protection
SMEP - Supports Supervisor Mode Execution Prevention
SMAP - Supports Supervisor Mode Access Prevention
PAGE1GB - Supports 1 GB large pages
PAE * Supports > 32-bit physical addresses
PAT * Supports Page Attribute Table
PSE * Supports 4 MB pages
PSE36 * Supports > 32-bit address 4 MB pages
PGE * Supports global bit in page tables
SS * Supports bus snooping for cache operations
VME * Supports Virtual-8086 mode
RDWRFSGSBASE - Supports direct GS/FS base access
FPU * Implements i387 floating point instructions
MMX * Supports MMX instruction set
MMXEXT - Implements AMD MMX extensions
3DNOW - Supports 3DNow! instructions
3DNOWEXT - Supports 3DNow! extension instructions
SSE * Supports Streaming SIMD Extensions
SSE2 * Supports Streaming SIMD Extensions 2
SSE3 * Supports Streaming SIMD Extensions 3
SSSE3 * Supports Supplemental SIMD Extensions 3
SSE4a - Supports Streaming SIMDR Extensions 4a
SSE4.1 * Supports Streaming SIMD Extensions 4.1
SSE4.2 * Supports Streaming SIMD Extensions 4.2
AES * Supports AES extensions
AVX * Supports AVX instruction extensions
FMA - Supports FMA extensions using YMM state
MSR * Implements RDMSR/WRMSR instructions
MTRR * Supports Memory Type Range Registers
XSAVE * Supports XSAVE/XRSTOR instructions
OSXSAVE * Supports XSETBV/XGETBV instructions
RDRAND - Supports RDRAND instruction
RDSEED - Supports RDSEED instruction
CMOV * Supports CMOVcc instruction
CLFSH * Supports CLFLUSH instruction
CX8 * Supports compare and exchange 8-byte instructions
CX16 * Supports CMPXCHG16B instruction
BMI1 - Supports bit manipulation extensions 1
BMI2 - Supports bit manipulation extensions 2
ADX - Supports ADCX/ADOX instructions
DCA - Supports prefetch from memory-mapped device
F16C - Supports half-precision instruction
FXSR * Supports FXSAVE/FXSTOR instructions
FFXSR - Supports optimized FXSAVE/FSRSTOR instruction
MONITOR * Supports MONITOR and MWAIT instructions
MOVBE - Supports MOVBE instruction
ERMSB - Supports Enhanced REP MOVSB/STOSB
PCLMULDQ * Supports PCLMULDQ instruction

Compiler Flags und Gleitkommazahlen

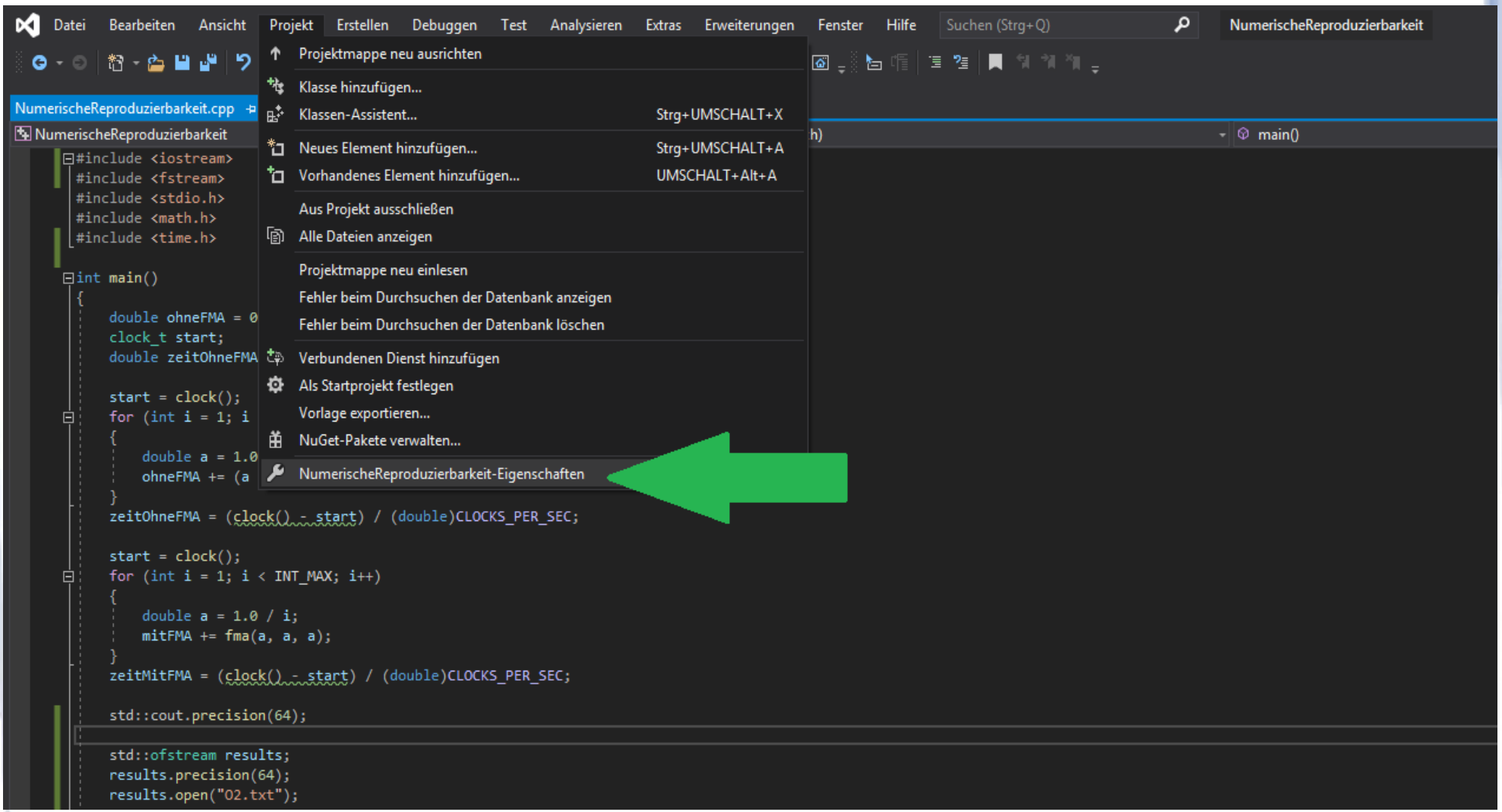
- Flags sind Anweisungen für Codeübersetzung
  - Begrenzungen
    - ffloat-store
  - Warnungen
    - Wfatal-errors
  - Optimierung
    - O2

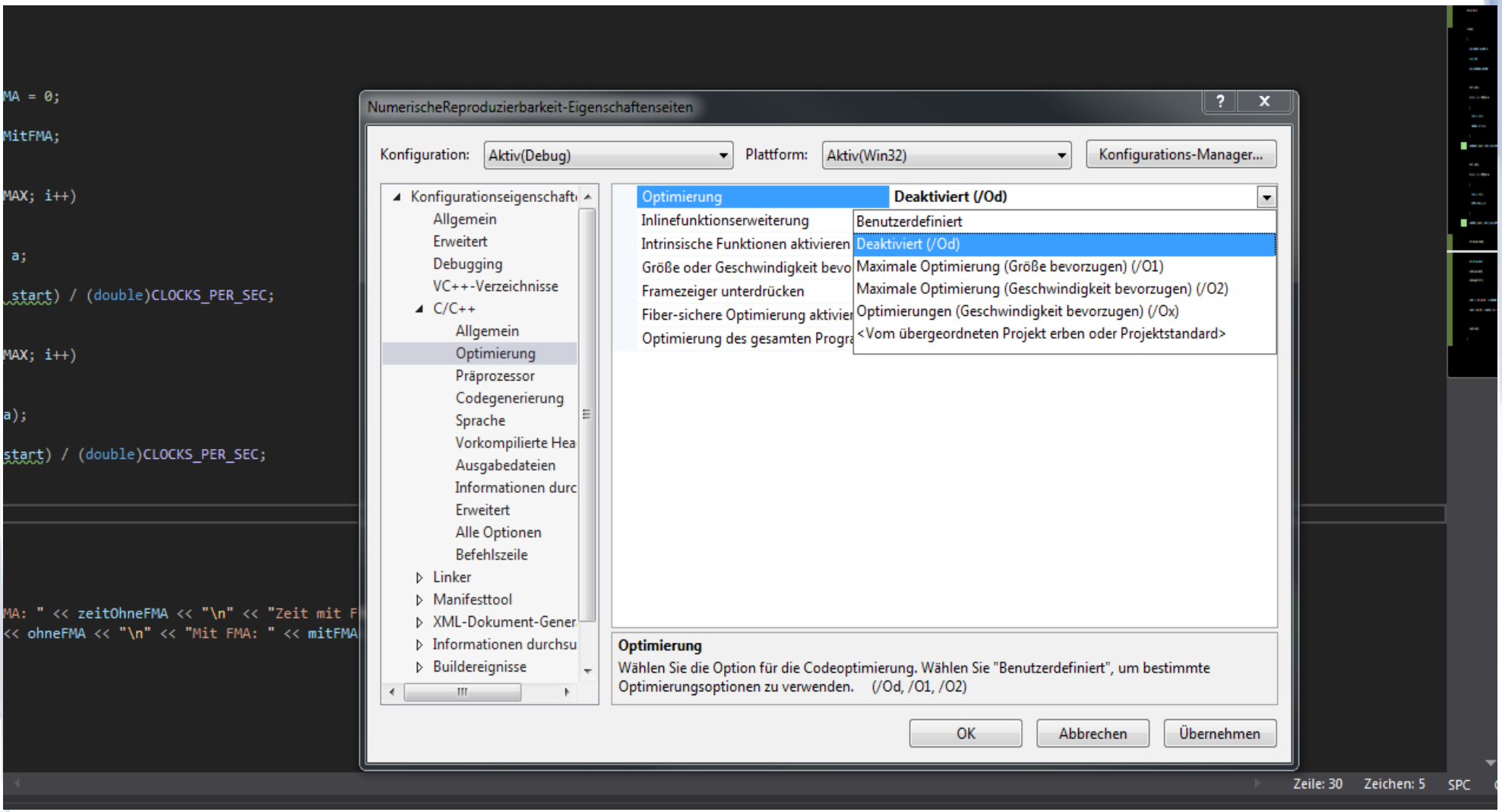
Optimierungsstufen

- O0
- O1
  - Minimierung der Codelänge
  - Niedrige Kompilierungszeit
- O2
  - Verbesserte Ausführungsgeschwindigkeit
  - Mehr Kompilierungszeit
- O3
  - Steigerung von O2



https://gi.de/informatiklexikon/reproduzierbarkeit


Datei
Bearbeiten
Ansicht
Projekt
Erstellen
Debuggen
Test
Analysieren
Extras
Erweiterungen
Fenster
Hilfe
Suchen (Strg+Q)
NumerischeReproduzierbarkeit
Projektmappe neu ausrichten
Klasse hinzufügen...
Klassen-Assistent... Strg+UMSCHALT+X
Neues Element hinzufügen... Strg+UMSCHALT+A
Vorhandenes Element hinzufügen... UMSCHALT+Alt+A
Aus Projekt ausschließen
Alle Dateien anzeigen
Projektmappe neu einlesen
Fehler beim Durchsuchen der Datenbank anzeigen
Fehler beim Durchsuchen der Datenbank löschen
Verbundenen Dienst hinzufügen
Als Startprojekt festlegen
Vorlage exportieren...
NuGet-Pakete verwalten...
NumerischeReproduzierbarkeit-Eigenschaften
NumerischeReproduzierbarkeit.cpp
NumerischeReproduzierbarkeit
main()
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <math.h>
#include <time.h>
int main()
{
double ohneFMA = 0
clock_t start;
double zeitOhneFMA
start = clock();
for (int i = 1; i
{
double a = 1.0
ohneFMA += (a
}
zeitOhneFMA = (clock() - start) / (double)CLOCKS_PER_SEC;
start = clock();
for (int i = 1; i < INT_MAX; i++)
{
double a = 1.0 / i;
mitFMA += fma(a, a, a);
}
zeitMitFMA = (clock() - start) / (double)CLOCKS_PER_SEC;
std::cout.precision(64);
std::ofstream results;
results.precision(64);
results.open("02.txt");


NumerischeReproduzierbarkeit-Eigenschaftenseiten
Konfiguration: Aktiv(Debug)
Plattform: Aktiv(Win32)
Konfigurations-Manager...
Konfigurationseigenschaft
Allgemein
Erweitert
Debugging
VC++-Verzeichnisse
C/C++
Allgemein
Optimierung
Präprozessor
Codegenerierung
Sprache
Vorkompilierte Hea
Ausgabedateien
Informationen durc
Erweitert
Alle Optionen
Befehlszeile
Linker
Manifesttool
XML-Dokument-Gener
Informationen durchsu
Buildereignisse
Optimierung
Deaktiviert (/Od)
Inlinefunktionserweiterung
Intrinsische Funktionen aktivieren
Größe oder Geschwindigkeit bevo
Framezeiger unterdrücken
Fiber-sichere Optimierung aktivier
Optimierung des gesamten Progra
Benutzerdefiniert
Deaktiviert (/Od)
Maximale Optimierung (Größe bevorzugen) (/O1)
Maximale Optimierung (Geschwindigkeit bevorzugen) (/O2)
Optimierungen (Geschwindigkeit bevorzugen) (/Ox)
<Vom übergeordneten Projekt erben oder Projektstandard>
Optimierung
Wählen Sie die Option für die Codeoptimierung. Wählen Sie "Benutzerdefiniert", um bestimmte Optimierungsoptionen zu verwenden. (/Od, /O1, /O2)
OK
Abbrechen
Übernehmen
Zeile: 30
Zeichen: 5
SPC

Zwischenfazit

- Rundungsmodi erleichtern Reproduzierbarkeit
  - Spezifizierte Abweichungen/Rundungen werden übernommen
- Fehlende Assoziativität verhindert Reproduzierbarkeit
- Implementierung der Rundungsmodi ist kompliziert
  - Unterschiede zwischen Sprachen und Hardware
  - Auswahl des Rundungsmodus

- Problem:

  Selbst reproduzierbare Lösungen sind nicht exakt

  - Richtung der Abweichung unbekannt und nicht nachvollziehbar

- Lösungsansatz:

  Daten(-intervalle) zu Lösungsintervallen verrechnen

Intervalle

- Inklusionsprinzip
  - Intervall umfasst exaktes Ergebnis
  - Genaue Anwendung der Arithmetik notwendig
- Verschiedene Darstellungsformen
  - [unterer Endpunkt, oberer Endpunkt]
  - Mittelpunkt ± r
  - …

Intervallarithmetik

- Kommutativ
- Assoziativ

- Keine Distributivität
- Keine additiven / multiplikativen Inversen

Rechenregeln

- Addition:
  - [1, 2] + [3, 4] = [(1+3), (2+4)] = [4, 6]
- Subtraktion:
  - [1, 2] – [3, 4] = [(1-4), (2-3)] = [-3, -1]

Rechenregeln

- Multiplikation:
  - [1, 2] * [3, 4] = [min(S), max(S)] = [3, 8]
    - S = {1*3, 1*4, 2*3, 2*4}
- Division:
  - [1, 2] / [3, 4] = [1, 2] * (1 / [3, 4]) = [(1/4), (2/4)]
  - (1 / [3, 4]) = [(1/4), (1/3)]

Abhängigkeitsproblem

- Intervalle nicht mit sich selbst verrechnen!

- $f(x) = x^2$
  - f([-1, 2]) = [0, 4]
  - [-1, 2] * [-1, 2] = [-2, 4]

- Überschätzung

Genauigkeit

- Überschätzung nicht nachvollziehbar
  - Rundungen
  - Abhängigkeiten
  - …
- Inklusionsprinzip wird eingehalten
  - Gerichtetes Runden
- Iterative Algorithmen ermöglichen Präzision

Fazit

- Intervallberechnungen sind aufwendig
  - Fehlende Assoziativität von Gleitkommazahlen
- Gleitkommazahlen gut für Abschätzungen
- Intervalle gut für Reproduzierbarkeit

Fazit

- Gleitkommazahlen sind nicht assoziativ!
- Dokumentationen genau lesen!
- Erwartungen anpassen

Quellen

- https://digitalcommons.csbsju.edu/cgi/viewcontent.cgi?article=1028&context=hon
- https://docs.oracle.com/cd/E19422-01/819-3693/819-3693.pdf
- https://floating-point-gui.de
- https://interval.louisiana.edu/kearfott.html
- http://www.mscs.mu.edu/%7Egeorgec/IFAQ/casares1.html
- http://www.mscs.mu.edu/%7Egeorgec/IFAQ/index.html
- http://www.cs.utep.edu/interval-comp/
- https://books.google.de/books?id=JTc4XdXFnQIC&pg=PA34&lpg=PA34&dq=inte
- https://docs.microsoft.com/de-de/sysinternals/downloads/coreinfo
- https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html
- https://www.youtube.com/watch?v=SLHnQuAPLsI

- Nathalie Revol und Phillippe Théveny, „Numerical Reproducibility and Parallel Computations: Issues for Integral Algorithms“, IEEE transactions on computers, Vol 63, No. 8, August 2014
- R. Moore, R.B. Kearfott, und M. J. Cloud, „Introduction to Interval Analysis“, Philadelphia, PA, USA: SIAM, 2009 - http://www-sbras.nsc.ru/interval/Library/InteBooks/IntroIntervAn.pdf
- https://gi.de/informatiklexikon/reproduzierbarkeit

Danke für eure Aufmerksamkeit!