Mixed Precision Runge-Kutta methods using Half Precision on the A64FX Processor

Ben Burnett [1] Sigal Gottlieb [1] Zachary J. Grant [2]
Alfa Heryudono [1]

[1]University of Massachusetts Dartmouth

[2]Oak Ridge National Laboratory

3th Annual Boston Area Architecture Workshop
January 28th, 2022

Table of Contents

Table of Contents

No Free Lunch

"Nothing is acquired for free, and necessarily must cost us some thing" -Epictetus

No Free Lunch

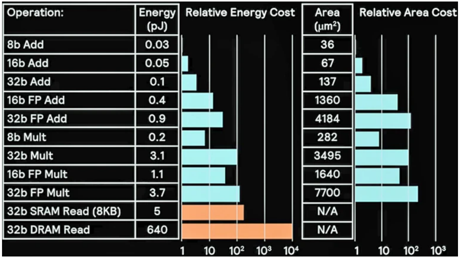
"Nothing is acquired for free, and necessarily must cost us some thing"
-Epictetus

IEEE 754 Floating Point Standard				
Name	Significand Bits	Exponent Bits	Exponent Min	Exponent Max
Half	11	5	-14	+15
Single	24	8	-126	+127
Double	53	11	-1022	+1023
Quadruple	113	15	-16382	+16383

No Free Lunch

"Nothing is acquired for free, and necessarily must cost us some thing"
-Epictetus

IEEE 754 Floating Point Standard				
Name	Significand Bits	Exponent Bits	Exponent Min	Exponent Max
Half	11	5	-14	+15
Single	24	8	-126	+127
Double	53	11	-1022	+1023
Quadruple	113	15	-16382	+16383



Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

Figure: Mark Horowitz, ISSCC presentation 2014

Table of Contents

Motivating Example: The Van der Pol Equation

For the General ODE

$$\frac{d}{dt}u = F(u)$$

solve the Van der Pol system

$$\begin{aligned} u_1' &= u_2 \\ u_2' &= u_2(1 - u_1^2) - u_1 \end{aligned}$$

Motivating Example: The Van der Pol Equation

For the General ODE

$$\frac{d}{dt}u = F(u)$$

solve the Van der Pol system

$$\begin{aligned} u_1' &= u_2 \\ u_2' &= u_2(1 - u_1^2) - u_1 \end{aligned}$$

using the Implicit Midpoint

$$y^{(1)} = u^n + \frac{\Delta t}{2} F\left(y^{(1)}\right)$$

$$u^{n+1} = u^n + \Delta t F\left(y^{(1)}\right)$$

Motivating Example: The Van der Pol Equation

For the General ODE

$$\frac{d}{dt}u = F(u)$$

solve the Van der Pol system

$$\begin{aligned} u_1' &= u_2 \\ u_2' &= u_2(1 - u_1^2) - u_1 \end{aligned}$$

using the Implicit Midpoint

$$y^{(1)} = u^n + \frac{\Delta t}{2} F\left(y^{(1)}\right)$$

$$u^{n+1} = u^n + \Delta t F\left(y^{(1)}\right)$$

Newton-Raphson Method

$$g(y^{(1)}) = 0 = u^n + \frac{\Delta t}{2} F\left(y^{(1)}\right) - y^{(1)}$$

To solve, use an expensive iteration until a tolerance is reached

$$y_{i+1}^{(1)} = y_i^{(1)} - J_i^{-1} g(y_i^{(1)})$$

Using Mixed Precision

Mixed-Precision Implicit Midpoint Method

$$y_\epsilon^{(1)} = u^n + \frac{\Delta t}{2} F^\epsilon\left(y_\epsilon^{(1)}\right)$$

$$u^{n+1} = u^n + \Delta t F\left(y_\epsilon^{(1)}\right)$$

Using Mixed Precision

Mixed-Precision Implicit Midpoint Method

$$y_{\epsilon}^{(1)} = u^n + \frac{\Delta t}{2} F^{\epsilon}\left(y_{\epsilon}^{(1)}\right)$$

$$u^{n+1} = u^n + \Delta t F\left(y_{\epsilon}^{(1)}\right)$$

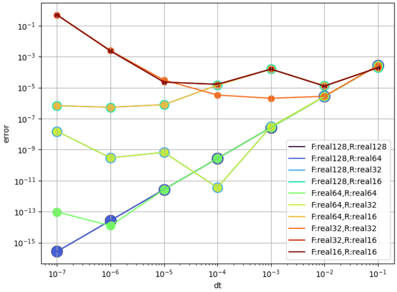


Figure: Errors from mixed-precision implicit midpoint rule

Errors calculated in extended precision regardless of computations base precision from a reference solution done in quadruple precision using a 4th order RK method with smaller Δt

Using Mixed Precision

Mixed-Precision Implicit Midpoint Method

$$y_\epsilon^{(1)} = u^n + \frac{\Delta t}{2} F^\epsilon\left(y_\epsilon^{(1)}\right)$$

$$u^{n+1} = u^n + \Delta t F\left(y_\epsilon^{(1)}\right)$$

Using Mixed Precision

Mixed-Precision Implicit Midpoint Method

$$y_\epsilon^{(1)} = u^n + \frac{\Delta t}{2} F^\epsilon\left(y_\epsilon^{(1)}\right)$$
$$u^{n+1} = u^n + \Delta t F\left(y_\epsilon^{(1)}\right)$$

Corrected Mixed-Precision Implicit Midpoint Method
for $k = 1, ..., p-1$

$$\begin{aligned} y_{[0]}^{(1)} &= u^n + \frac{\Delta t}{2} F^\epsilon\left(y_{[0]}^{(1)}\right) \\ y_{[k]}^{(1)} &= u^n + \frac{\Delta t}{2} F\left(y_{[k-1]}^{(1)}\right) \\ u^{n+1} &= u^n + \Delta t F\left(y_{[p-1]}^{(1)}\right) \end{aligned}$$

Performance of Mixed-Precision Methods

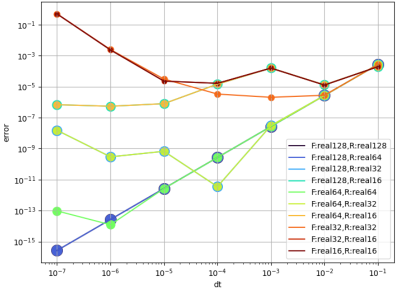


Figure: No corrections

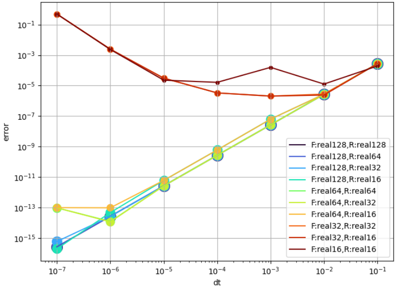


Figure: One correction

Performance of Mixed-Precision Methods

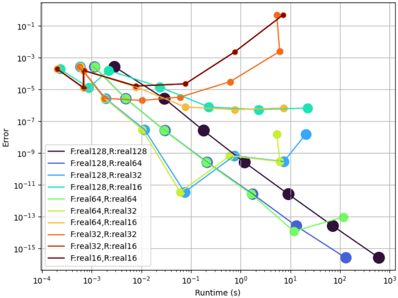


Figure: No corrections

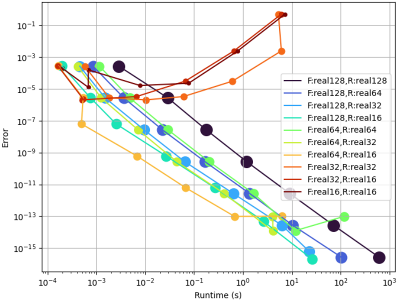


Figure: One correction

Power Consumption of Mixed-Precision Methods

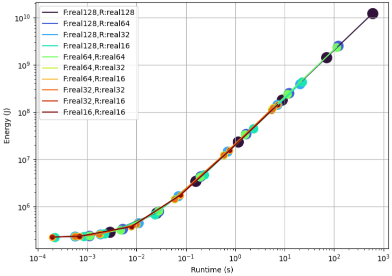


Figure: Energy (J) Consumption vs Run Time of Solver

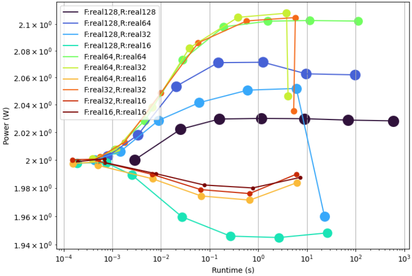


Figure: Power (W) Consumption vs Run Time of Solver

Power and energy consumption was collected from the A64FX processors energy counters using `perf`

Table of Contents

Conclusions

Key Takeaway:

Potential to save an appreciable amount of time while still being able to maintain high accuracy.

Conclusions

Key Takeaway:

Potential to save an appreciable amount of time while still being able to maintain high accuracy.

Further Work:

- More problems (Are problems that require other methods to solve the implicit stage more likely to benefit?)
- More methods (Do solvers with more stages benefit more/less?)

Conclusions

Key Takeaway:

Potential to save an appreciable amount of time while still being able to maintain high accuracy.

Further Work:

- More problems (Are problems that require other methods to solve the implicit stage more likely to benefit?)
- More methods (Do solvers with more stages benefit more/less?)

Thank you for attending! Any questions?