Analysing and bounding error sources in numerical neural network simulations

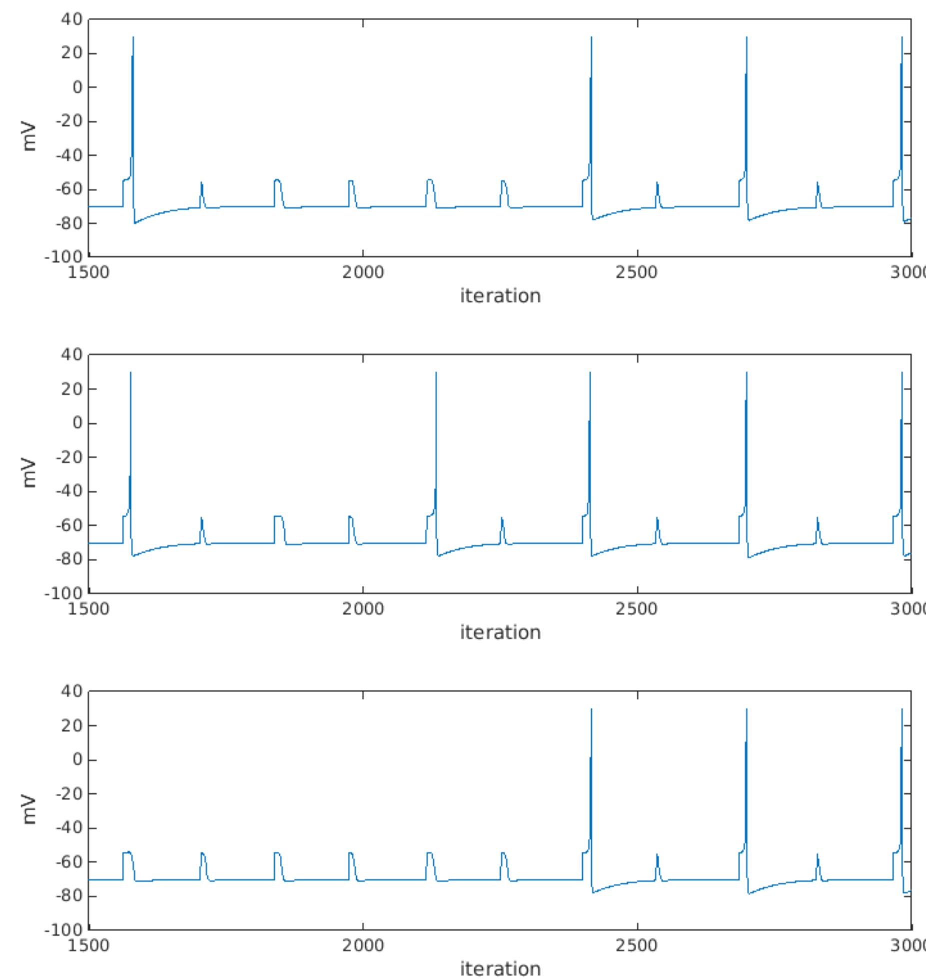
James Paul Turner

University of Sussex

School of Engineering and Informatics,
Centre for Computational Neuroscience and Robotics.

Numerical Error in Neural Networks

- **Reproducibility is desirable, but tricky**
- **Different results with different compilers and architecture**
- **Even different in the same environment**
- **Need a way to measure *how* reproducible a simulation is…**



Numerical Error in Neural Networks

Rounding error: $$x \, op \, y = \left(x \; \widehat{op} \; y \right)\left(1+\delta\right)$$

Nondeterminism: $$\left(\left(x \, \widehat{op} \; y \right) \; \widehat{op} \; z \right) \neq \left(x \, \widehat{op} \left(y \, \widehat{op} \, z \right)\right)$$

Truncation error: $$x_{t+h} = x_t + \left.\frac{dx}{dt}\right|_t h + O\left(h^2\right)$$

- **Lots of nondeterminism in parallel hardware**
- **Errors quickly amplified in unstable systems**

Interval Arithmetic (IA)

- **Replace variables and operators with IA equivalents:**

$$x \in \bar{x}, y \in \bar{y}, z \in \bar{z}$$
$$\bar{x} = [x_{lo}, x_{hi}]$$
$$\bar{y} = [y_{lo}, y_{hi}]$$
$$\bar{z} = \bar{x} - \bar{y} = [(x_{lo} - y_{hi}), (x_{hi} - y_{lo})]$$

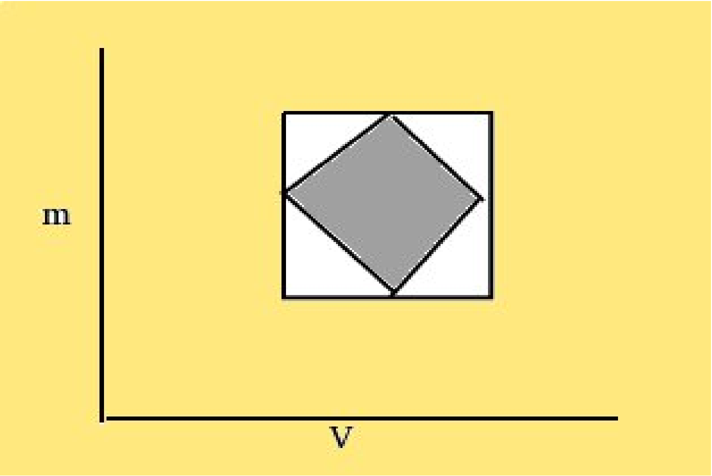
- **Cannot track correlations:**

$$p - p = 0$$
$$\bar{p} = [1, 2]$$
$$\bar{p} - \bar{p} = [(1-2), (2-1)] = [-1, 1]$$

- **Interval 'explosion' is an issue in long chained computations, such as numerical integration**

Interval Arithmetic (IA)

- **The ‘wrapping effect’**



- **IA tells us the axis-aligned white box is the reachable region**
- **If *m* and *V* are correlated, the real reachable region might in fact be the grey diamond**

Affine Arithmetic (AA)

- **Solution: encode correlations within representation**
- **Intervals are encoded as first-order polynomials with a centre: x_0, and n deviation terms: $x_i \varepsilon_i$**

$$\hat{x} = x_0 + x_1 \varepsilon_1 + \ldots + x_n \varepsilon_n$$

- **All ε_i are unknown, shared amongst all affine intervals and represent the uncertainty from an error source**

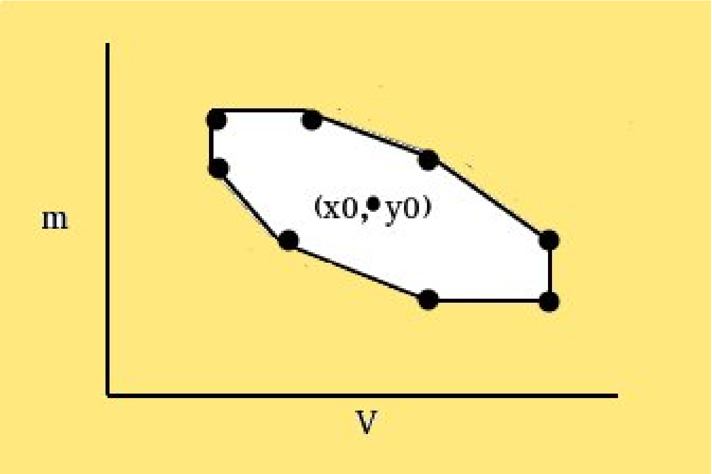
$$\varepsilon_i \in [-1, 1]$$

- **x and y are correlated iff:**

$$\exists k > 0 : |x_k| \neq 0 \wedge y_k \neq 0$$

Affine Arithmetic (AA)

- **A convex hull 'wraps' the reachable region better**
- **Deviation terms correspond to edges**



- **Numerical error is placed in a new deviation term $x_k \varepsilon_k$**

$$\hat{x} = x_0 + x_1 \varepsilon_1 + \ldots + x_n \varepsilon_n + x_k \varepsilon_k$$

- **Nonlinear operations are approximated linearly (Chebyshev), and linearisation error is also added to $x_k \varepsilon_k$**

Affine Arithmetic (AA)

affine_1 (z, x, α, γ, δ)

$$z = (\alpha x_0 + \gamma)$$
$$+ (\alpha x_1)\varepsilon_1 + \ldots + (\alpha x_n)\varepsilon_n$$
$$+ (\delta)\varepsilon_{n+1}$$

affine_2 (z, x, y, α, β, γ, δ)

$$z = (\alpha x_0 + \beta y_0 + \gamma)$$
$$+ (\alpha x_1 + \beta y_1)\varepsilon_1 + \ldots + (\alpha x_n + \beta y_n)\varepsilon_n$$
$$+ (\delta)\varepsilon_{n+1}$$

Affine Arithmetic (AA)

exp_cheb (z, x)

$\alpha = (e^{x.hi} - e^{x.lo}) / (x.hi - x.lo)$

$d_a = e^{x.lo} - \alpha x.lo$

$d_b = e^{x.hi} - \alpha x.hi$

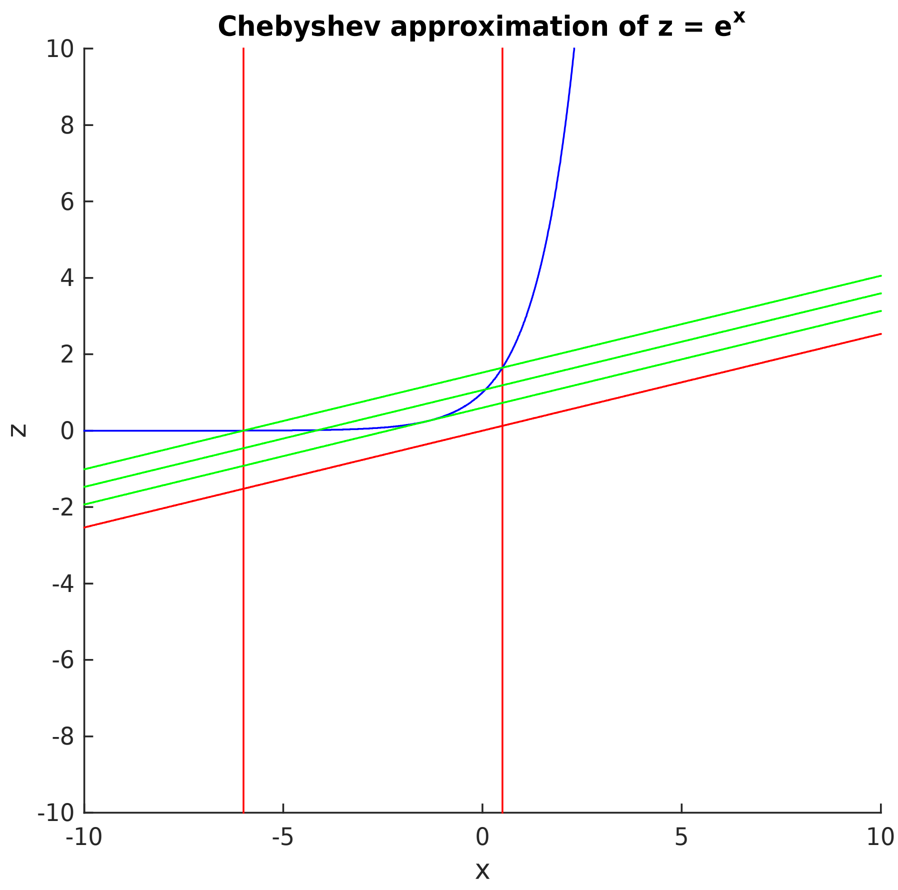
$d_{min} = e^{u} - \alpha u$

$d_{max} = \max(d_a, d_b)$

$\gamma = \text{mid}([d_{min}, d_{max}])$

$\delta = \text{rad}([d_{min}, d_{max}])$

affine_1(z, x, α, γ, δ)



The MPFA Library

- **Written in C**
- **Open Source (LGPLv3 license)**
- **Based on the GNU MPFR Library**
- **Link as a shared or static lib**
- **Uses the GNU Autotools, for simpler compilation, installation and portability**

  ./configure && make && sudo make install
- **Beta release very soon!**

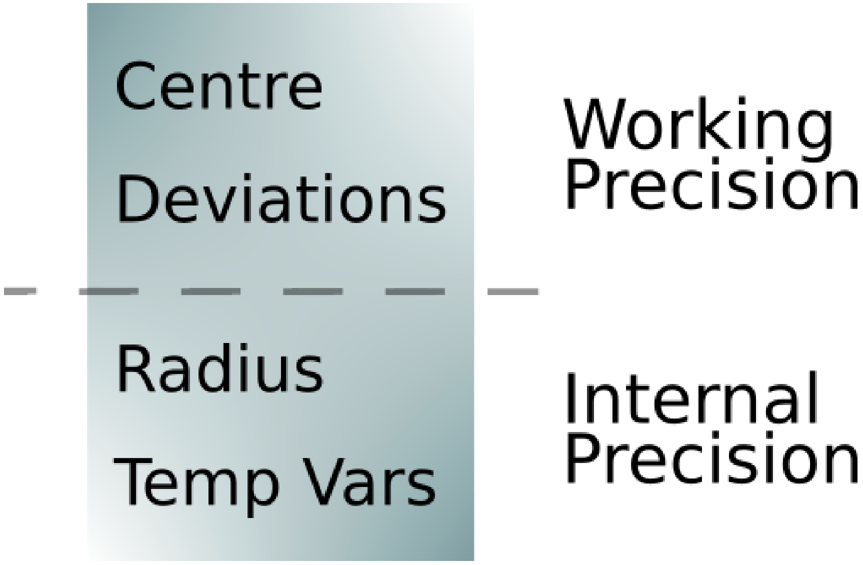
  Available: www.github.com/jamesturner246/mpfa

The MPFA Library

- **Benefits of MPFR**
  - Results guaranteed to be correctly rounded
  - Support for IEEE-754-2008 rounding modes
  - Precision is adjustable at runtime
  - No opaque compiler optimisation
  - Extensive testing
  - All the above, for **all** functions

The MPFA Library

- **Potentially lots of internal rounding error**
- **Centre and deviation terms are computed in working precision**
- **Intermediate variables and radius are computed in higher internal precision**



The MPFA Library

- **Many new deviation terms: 1 per operation**
- **Eventually need 'garbage collection'**

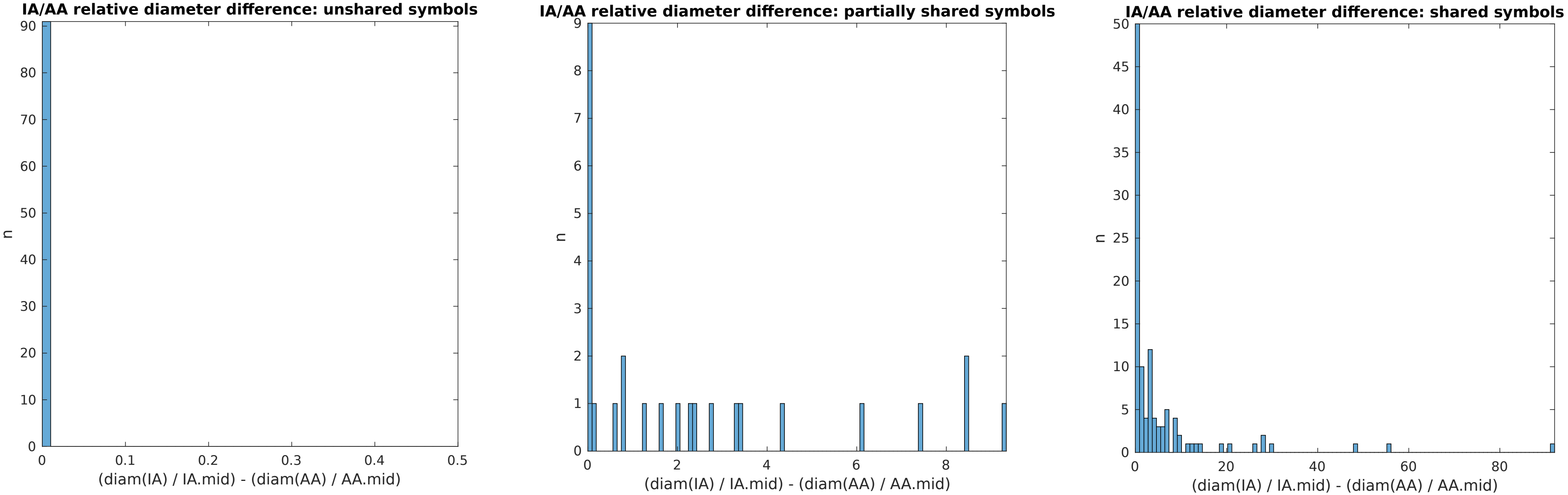
mpfa_condense_last_n (mpfa_t z, unsigned n)

- Condense the last *n* deviation terms
- Lossless (when used correctly)

mpfa_condense_small (mpfa_t z, double fraction)

- Condense terms smaller than some fraction of radius
- Lossy (some correlation information is lost)

The MPFA Library



mpfa_add and mpfi_add, 1,000,000 tests, 100 bins

Morris-Lecar Neuron Model

$$C\frac{dV}{dt}=I-g_L\left(V-V_L\right)-g_{Ca}M_{ss}\left(V\right)\left(V-V_{Ca}\right)-g_K N\left(V-V_K\right)$$

$$\frac{dN}{dt}=\frac{N_{ss}\left(V\right)-N}{\tau_N\left(V\right)}$$

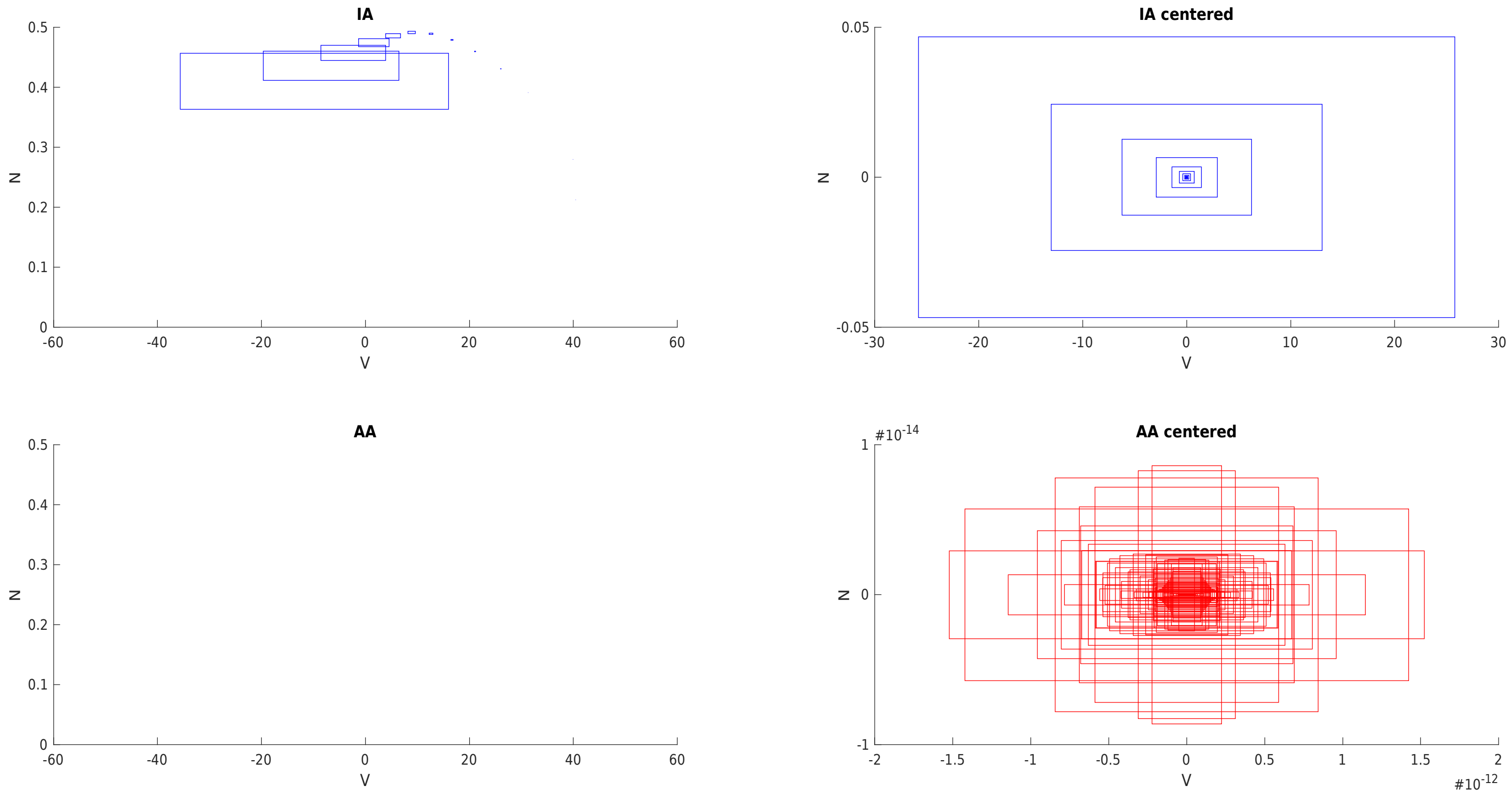
$$M_{ss}\left(V\right)=\frac{1}{2}\left(1+\tanh\left(\frac{V-V_1}{V_2}\right)\right)$$

$$N_{ss}\left(V\right)=\frac{1}{2}\left(1+\tanh\left(\frac{V-V_3}{V_4}\right)\right)$$

$$\tau_N\left(V\right)=1/\left(\phi\cosh\left(\frac{V-v_3}{2V_4}\right)\right)$$

Morris-Lecar Neuron Model

IA (top) and AA (bot). Iterations 1 to 83.



Thanks!

The MPFA Library

An arbitrary-precision affine arithmetic library based on GNU MPFR.

Open source permissive license (LGPLv3).

Download, contribute and give feedback at:

www.github.com/jamesturner246/mpfa