Numerics: from instabilities to uncertainties

Nikolas Kauer

Centre for Particle Physics

Royal Holloway, University of London

in the GOLEM group

Workshop on fixed order multi-leg automatic NLO calculations

University of Wuppertal

June 3, 2009

Outline

- Numerical (in)stability & loop calculations
- Strategy: Avoid - Detect - Remedy
- From instabilities to uncertainties
- Stochastic arithmetic/Monte Carlo arithmetic
- Unified treatment with physical uncertainties
- Summary

Numerical (in)stability

Numerical Analysis $\longrightarrow$ numerical algorithms $\longrightarrow$ numerical stability

Computer-based numerical evaluation: very powerful, but introduces approximations:

IEEE floating-point arithmetic: finite precision

$\varepsilon_{\text{mach}} = 2^{1-p}, p = 24, 53, 113$ for single, double, quad

$\longrightarrow$ Rounding, different **Modes** implemented: to nearest or directed

Problem: 1 rounding: small error ✓, 1 million roundings: still small error?

Operations order becomes significant:

$\boxed{1 + 10^{-20} - 1} \rightarrow \boxed{0.}$, but $\boxed{1 - 1 + 10^{-20}} \rightarrow \boxed{9.99999999999999995e - 21}$

$n = \frac{n}{g} + n - \frac{n}{g} \rightarrow \boxed{0.}$ if $g \lesssim \varepsilon_{\text{mach}}$

Algorithms that can be proven **not to** magnify approx. errors are called **num. stable**.

Goal: evaluate expressions that do not magnify approximation errors for relevant input values.

Loop amplitude evaluation

inverse Gram and other kinematical determinants

$\longrightarrow$ large cancellations can occur in critical phase space regions

$\longrightarrow$ numerical instabilities

Strategy: Avoid - Detect - Remedy

Avoidance

- optimise representation (math & code)
- minimize number of terms
- many competing methods, schools of thought
- symbolically cancel spurious small denominators
  $\longleftarrow$ inverse kinematical determinants

Detection

- numerically check known relations (OPP, ...)
  (if required subexpressions $\longrightarrow$ on the fly)
- re-evaluate with increased precision and compare (?)
- (compare numerical with analytically- known results)
- general numerical methods

Is a relations-based method sensitive to all instability sources?

Is there a tensor coefficient-scalar integral interplay for instabilities?

Remedies

Analytical remedies

- analytical identification of critical kinematical configurations
- expand integrand expression in critical regions about small parameters
- extrapolate integrand/integral into critical regions

Numerical remedies

- re-evaluate in quadruple (or higher) precision

Fortran **compiler** support for quad precision:
commercial compilers (Intel, Absoft) ✓, GNU/free compilers (gfortran, g95) not yet

Quadruple and arbitrary precision **libraries**:
LBL high precision software directory: ARPREC, QD, ... (Fortran/C++), also: GNU MPFR (C/C++)

software implementation $\longrightarrow$ runtime penalty (factor ~ 20)
if only used in **small fraction** of operations $\longrightarrow$ $\sim \mathcal{O}(1)$ longer overall runtime

1: use only for PS points where double precision fails, 2: use only in affected subexpressions

Validation

current best option: use different methods, compare results $\longrightarrow$ "error estimate"

advantageous: modular packages that can be interfaced easily

Current practice

BlackHat

Berger, Bern, Dixon, Febres Cordero, Forde, Ita, Kosower, Maitre

FormCalc

Hahn, Rauch, ...

GOLEM

Binoth, Guffanti, Guillet, Heinrich, Karg, Kauer, Reiter, Reuter

HELAC/CutTools

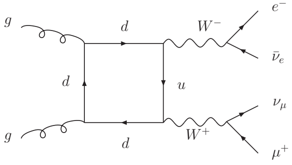
Bevilacqua, Cafarella, Czakon, van Hameren, Kanaki, Ossola, Papadopoulos, Pittau, Worek, ...

Rocket

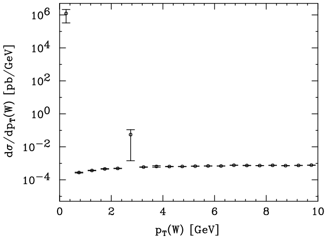
Ellis, Giele, Kunszt, Melnikov, Zanderighi

$\longrightarrow$ talks by Hahn, Maitre, Papadopoulos, Reiter, Zanderighi

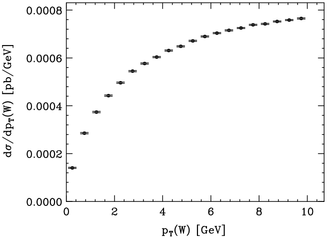
Quad precision: stability guaranteed?



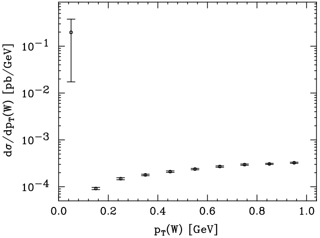
$gg \to W^+W^- \to$ leptons: box



no cuts, $\sigma_{tot} = 60.2$ fb, double precision
(instability at $p_{TW} \approx 3$ GeV $\to$ technical cuts insufficient)

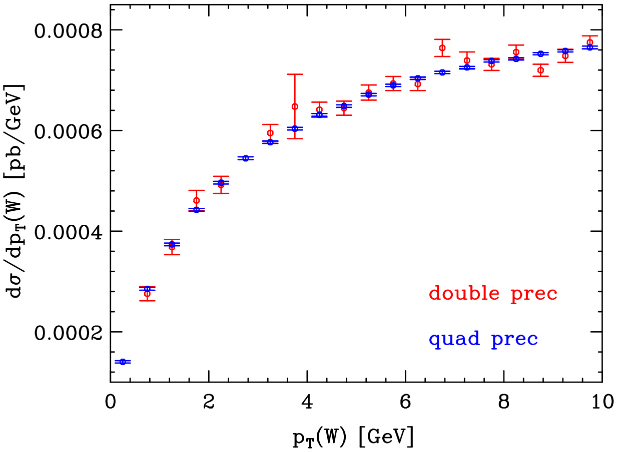


no cuts, $\sigma_{tot} = 60.2$ fb, quad precision



$p_T(W) < 1$ GeV, $\sigma \approx 0.2$ fb, quad precision

Instabilities can occur without being catastrophic



General approaches to determine result accuracy/uncertainty

$\longrightarrow$ additional tools to validate algorithms used by automatic packages

Interval arithmetic

- perform arithmetic operations on intervals rather than numbers
- idea: $x \longrightarrow [x_{\text{min}}, x_{\text{max}}]$ or $[x - \Delta x, x + \Delta x]$ etc.
- accurate ranges, but dependency issue for complicated expressions

"Crank three times" (JS Denker)

- calculate result multiple times with perturbed input values (min, max)
- e.g. external momenta $\pm \sim \varepsilon_{\text{mach}}$

Stochastic arithmetic/Monte Carlo arithmetic

- replace computer's deterministic arithmetic by stochastic arithmetic
- each operation is performed n times before the next operation is executed
- propagate round-off error differently each time

Scott, Jezequel, Denis, Chesneaux, CPC 176 (2007) 507

Parker, Pierce, Eggert, Computation in Science and Engineering 2 (2000) 58

Interval arithmetic

Accurate range determination of elementary arithmetic operations and simple functions

Dependency issue for complicated expressions

If the same input interval ($\leftarrow$ value) occurs several times in the expression that is evaluated and each occurrence is taken independently, this can lead to an undesirable expansion of the resulting intervals.

A toy example: $y := f(x) = x^2 + x$, $x \in [-1, 1] \quad \Rightarrow \quad y \in [-0.25, 2]$
independent evaluation yields $[-1, 1]^2 + [-1, 1] = [0, 1] + [-1, 1] = [-1, 2]$

Practical solutions for lengthy expressions?

Local expert: Prof. Walter Krämer (Informatics Group, University of Wuppertal)
Research interests:

- tools to automatise error estimation
- mathematical functions with safe error bounds
- numerics with result verification

Implementations: *Extensions for Scientific Computation* XSC (Wuppertal-Karlsruhe), PROFIL/BIAS, Boost (C++ template), ...

CADNA: Numerical “health check” for scientific codes

Probing round-off error propagation with rounding modes

Four modes defined in IEEE FP arithmetic standard: **nearest, zero, $-$inf, $+$inf**

A toy example: `x = 1.; y = 1.e-20;` (single precision), compute:
`z1 = x - y; z1 = z1 - x; z2 = y - x; z2 = z2 + x;`

	nearest	$-$inf	$+$inf	zero
z1	0.0000000000e+00	-5.9604644775e-08	0.0000000000e+00	-5.9604644775e-08
z2	0.0000000000e+00	-0.0000000000e+00	5.9604644775e-08	5.9604644775e-08

CADNA (Control of Accuracy and Debugging for Numerical Applications) goals:

- report gradual and catastrophic loss of precision (due to round-off error propagation)
- be of acceptable efficiency
- be non-invasive to the source code

CADNA/CESTAC method: stochastic triples $\longrightarrow$ (mean, std. dev.), round using mode $-$inf or $+$inf, randomly round with probability 0.5 to obtain 1st and 2nd value, obtain 3rd value with mode not used for 2nd.

Recommended reading: Scott, Jezequel, Denis, Chesneaux, CPC 176 (2007) 507

CADNA library (Fortran): www.lip6.fr/cadna

Unified treatment with physical uncertainties

Consider CADNA stochastic triples (mean, std. dev.) as special case of input parameter uncertainty

Apply Stochastic arithmetic/Monte Carlo arithmetic approach to account for

- round-off error
- input parameter uncertainty (couplings, masses, ...)
- scale uncertainties (sample μ_R, μ_R independently in $[\mu_{\text{min}}, \mu_{\text{max}}]$)
- PDF uncertainties (sample PDF eigenvector sets)
- experimental uncertainties (detector effects)

On-the-fly separation of MC integration error from physical uncertainty?

Summary

- Numerical (in)stability & loop calculations
- Strategy: Avoid - Detect - Remedy
- From instabilities to uncertainties
- Stochastic arithmetic/Monte Carlo arithmetic
- Unified treatment with physical uncertainties