A fast, simple, and naturally machine-precision algorithm for calculating both symmetric and asymmetric $M_{T2}$, for any physical inputs.

Christopher G. Lester$^a$ and Benjamin Nachman$^b$

$^a$Department of Physics, University of Cambridge  
$^b$Department of Physics, Stanford University  
E-mail: lester@hep.phy.cam.ac.uk, bnachman@cern.ch

ABSTRACT: This document describes a transverse-mass calculation algorithm that has better numerical stability, and therefore accuracy, than the fastest existing implementations. The new algorithm naturally permits computation of $M_{T2}$ to machine-precision for any valid set of inputs. In addition to being more accurate than existing fast calculators, the new implementation is arguably simpler to understand, comprises fewer lines of active code, and provides the first fast machine-precision asymmetric-$M_{T2}$ calculator known to the authors.

1 Introduction

For the purposes of this document, we will define the kinematic variable $M_{T2}$ in the following way:

$$M_{T2}(m_s, \vec{s}, m_t, \vec{t}, \vec{p}, \chi_1, \chi_2) = \min_{\vec{p}, \vec{q} \text{ s.t. } \vec{p} + \vec{q} = \vec{p}} \{ \max [M_T(m_s, \vec{s}, \chi_1, \vec{p}), M_T(m_t, \vec{t}, \chi_2, \vec{q})] \}$$

where the transverse mass is given by

$$M_T(m, \vec{v}, \chi, \vec{p}) = \sqrt{m^2 + \chi^2 + 2\sqrt{|\vec{v}|^2 \chi^2 + |\vec{p}|^2 - 2\vec{v}.\vec{p}}}$$

in which $\vec{s}$, $\vec{t}$, $\vec{p}$, $\vec{q}$ and $\vec{p}$ are all real two-vectors, and the remaining quantities are real scalars which may all be assumed to be non-negative as they only enter through their squares. Until fairly recently, the majority of $M_{T2}$ usage in experimental literature concerned itself only with the so-called ‘symmetric’ case, $\chi_1 = \chi_2$, which is also the form in which $M_{T2}$ was first proposed [1]. However, there is a growing interest in the ‘asymmetric’ case, $\chi_1 \neq \chi_2$, [2, 3], which is a powerful tool for reducing asymmetric backgrounds to symmetric signal processes. For example, it was recently suggested to use asymmetric $M_{T2}$ to suppress the dominant two-lepton $tt$ background in searches for the supersymmetric top quark partners decaying into one charged lepton [4]. Such searches tend to require a large transverse mass, $M_T > M_W$, from the selected charged lepton and the missing momentum from the undetected neutrino (neutralinos in the signal can make $M_T \gg M_W$). This eliminates most of the one charged lepton $tt$ as $M_T \leq M_W$ and what is left is a two lepton $tt$ background with an unidentified electron/muon or hadronically de-
caying $\tau$ lepton. The natural choices for $\chi_1$ and $\chi_2$ are then $M_\nu$ and $M_W$. When the rest of the $t\bar{t}$ topology is correctly assigned, $M_{T_2}$ has an endpoint at $m_{stop}$ for the background. This property helped to extend the most stringent limits on $t \rightarrow t\chi_1^0$ production [5]. Asymmetric $M_{T_2}$ has been used to search for many more topologies, including cases in which the signal processes are expected to have low values of $M_{T_2}$ as in three body $t \rightarrow bW\chi_1^0$ decays. For these cases, the fact that a Jacobian peak pushes probability near the kinematic maximum of the $M_{T_2}$ distribution is crucial.

While it is a relatively trivial matter to construct an algorithm capable of evaluating $M_{T_2}$ by using an off-the-shelf numerical minimiser and the definition given above, such methods are usually very slow (since the minimum is often on a singular fold or crease) and error prone (as the minimum can be near infinity, or the crease may be steep sided but have a very shallow gradient along the fold). Therefore, ever since $M_{T_2}$’s creation there has been a need for fast and accurate methods for evaluating it, many of which can be found collated in [6]. The publication in 2008 of the ‘elliptic bisection’ method of Cheng and Han [7] caused a minor revolution in the world of $M_{T_2}$ algorithms. Their implementation (which we will refer to as ‘CH’) has since been regarded as the de-facto fastest and most reliable method of evaluating the symmetric form of the variable.

$M_{T_2}$ is, from a theoretical perspective, no harder to compute in the asymmetric case than in the symmetric. However, as a result of the historical accident that no-one was interested in asymmetric $M_{T_2}$ in 2008, the classic CH algorithm [7] is only capable of evaluating symmetric $M_{T_2}$. Consequently, physicists needing to evaluate asymmetric $M_{T_2}$ have had no option but to either: (1) attempt to compute the variable themselves with generic minimisers, exposing themselves to the dangers and severe performance penalties mentioned above, or (2) attempt to modify the CH algorithm to cope with the asymmetric case. Neither of these options have been achieved satisfactorily. Ample evidence that the latter route (CH modification) is much harder than one might naively expect may be found in Walker’s Table 1 [9] which shows that CH modifications which do not pay enough attention to the many areas of numerical instability that are carefully worked around in the CH algorithm, can easily lead to errors of hundreds of GeV in $M_{T_2}$ evaluations. It is possible to find ways around the majority of these issues, as Walker [9] demonstrates elegantly, but only at some considerable cost in algorithm complexity.\footnote{Note that [9] provides perl implementations of algorithms that can reliably evaluate $M_{T_2}$ in both the symmetric and asymmetric forms considered herein (and in some others). Though the choice of language makes these implementations slow, they would probably become much faster if re-written in a language. At present, being more than a factor of 1000 slower than CH, these implementations are not counted as ‘fast’ herein. Nonetheless, it will be interesting to see (1) how much faster they could be if reimplemented, and (2) whether they can also reach machine precision in all the cases where CH struggles, described later.}

Given the above, the primary goal of the work written up herein, was to produce a reliable fast $M_{T_2}$ calculator capable of serving the previously un-served needs of those wanting to use the asymmetric case. Rather than attempt to adapt the CH implementation itself, it was thought better to return to first principles, and ask why previous attempts to adapt CH had been so unsuccessful, complex or error prone.

2 The good and bad bits of bisec-

tion algorithms

A key insight within CH [7] was this: if it were possible to determine relatively quickly whether a ‘trial’ value $M$ were greater than or less than the actual value of $M_{T_2}$, without actually computing the actual value of $M_{T_2}$, then by repeatedly bisecting a finite interval known to enclose the true value, it would be possible to

2 The CH implementation is downloadable from [8] and is also distributed within [6].
determine \( M_{T2} \) in a cost effective manner.\(^3\) This is very important, and will be heavily relied upon in the new algorithm described herein.

A second important result of [7], and later works, was that the trial value, \( M \), could indeed be compared to the true value of \( M_{T2} \), without knowing the value of the latter. We will not describe the origin of this test in detail here as it is described elsewhere.\(^4\) However the main points we will need to understand are the following:

1. A trial value \( M \), together with \( m_s, \vec{s} \) and \( \chi_1 \), defines a (possibly empty) subset \( R_1(M) \) of points within \( \mathbb{R}^2 \).

2. A trial value \( M \), together with \( m_t, \vec{t}, \chi_2 \) and \( \vec{p} \), defines another (possibly empty) subset \( R_2(M) \) of points within \( \mathbb{R}^2 \).

3. Whenever \( R_i(M) \) (for \( i = 1 \) or \( 2 \)) is non-empty, it comprises either (i) a set of points within or on the boundary of a non-degenerate ellipse, or (ii) a set of points within or on the boundary of a non-degenerate parabola,\(^5\) or (iii) a set of points lying on a straight ‘ray’, starting at a point in \( \mathbb{R}^2 \) and radiating out to infinity in some direction (this being a limiting case of progressively narrower and narrower parabolas, under certain circumstances), or (iv) a single point (this being a limiting case of progressively smaller and smaller ellipses, under certain circumstances).

4. The two singular cases (iii) and (iv) described above occur if and only if \( M \) is equal to the kinematic minimum of the \( M_{T2} \) distribution, i.e. when \( M = \max[m_s + \chi_1, m_t + \chi_2] \).

5. \( R_1(M) \) (respectively \( R_2(M) \)) switches from elliptical to parabolic character (or from point to ray, in the singular case) when the mass \( m_s \) (respectively \( m_t \)) goes from non-zero to zero.

6. The value of \( M \) is greater than or equal to the true value of \( M_{T2} \) if and only if \( R_1(M) \cap R_2(M) \neq \emptyset \).

Item 6 above is the key beneficial feature that allows the bisection algorithms to work, and is used in the new algorithm we propose. Item 5, on the other hand, is the cause of most of the trouble in the existing implementations. It causes problems for three reasons which we will describe in turn: (a) co-ordinate instability, (b) special case proliferation, and (c) Lorentz-vector choices. We will describe the effects in terms of \( R_1(M) \) only, but the same arguments apply to \( R_2(M) \).

### 2.1 Co-ordinate instability

As \( m_s \to 0 \), but before 0 is actually reached, the ellipses bounding \( R_1(M) \) grow progressively longer and longer (and wider and wider at the same time) in a ‘one-sided’ manner — i.e. one of the two ends of the ellipse stays more or less where it is, eventually forming the pointy part of the parabola, while the other end of the ellipse grows and swells to infinite length and width in order to ‘open up’ into the infinite region that is the inside of the parabola. Because of this growth, the coordinates of the centre of the ellipse, and the lengths of its semi-major and semi-minor axes, all tend to infinity.\(^6\) Many existing implementations use these unstable coordinates internally, and have only limited protections against their being used in unstable regions. For example, the CH method is aware of this potential problem, and so first compares the magnitude of \( m_s \) to an ad-hoc threshold, and if \( m_s \) is dangerously light, flips over to treating \( m_s \) as if it were identically zero (even though it

\(^3\)The best reference for this is [9] which is complete in all respects. An accessible description of the main features can be found in [7], though it contains little concrete coverage of any of the special cases which are either not discussed or are left as exercises for the reader.

\(^4\)We define a point in \( \mathbb{R}^2 \) to lie ‘within’ a parabola if it lies on a line segment joining any two points on the parabola.

\(^5\)Given that all inputs to \( M_{T2} \) are finite quantities, such growth to infinity can only arise from dividing by quantities that become close to zero. It is astonishing how many such divisions can be found lurking in existing implementations. The new algorithm proposed here, has only one non-constant division, and it is provably non-dangerous. Much of the safety of the proposed implementation relies on this fact.
isn’t), since a custom massless algorithm knowing only about parabolae need not share the elliptical parametrisation. The need to flip at an arbitrary point, however, limits the ability to calculate precise values for any value of \( m_s \) below that threshold, and the precise positioning of the threshold might need considerable investigation and tuning, which complicates the development and use of the algorithm.

2.2 Special case proliferation

So far as the author can tell, when testing for the intersection in item 6 of section 2, all existing implementations examine some sort of discriminant, or Sturm Sequence, which is able to make a statement about the number or character of the points which lie on the two conics bounding \( R_1(M) \) and \( R_2(M) \). This is probably the largest single inefficiency such methods introduce. The basic problem is this: if the boundaries of \( R_1(M) \) and \( R_2(M) \) intersect then it is clear that \( R_1(M) \cap R_2(M) \neq \emptyset \) but the reverse is not true. For example, \( R_1(M) \cap R_2(M) \) may be non-empty without intersection of their boundaries when, say, \( R_1(M) \) is bounded by a small ellipse entirely contained within the interior of a larger ellipse bounding \( R_2(M) \). So if these discriminants determine that the boundaries do not intersect, additional special-case code then has to determine whether this is due to a trivial enclosure, or to non-intersection of the interiors. Such special-case code is complicated by the fact that different tests may be needed for parabola-nested-within-parabola, ellipse-nested-within-parabola, ray-nested-within-ellipse, and ellipse-nested-within-ellipse!

2.3 Lorentz-vector implementations

If the dangerous region occurs when \( m_s \) is close to zero, but not exactly so, why should we care? We don’t expect to measure non-zero jet masses to zero, but not exactly so, why should we care?

The principle innovation in the new algorithm, is that it finds a new way of approaching and performing the intersection test in item 6 of section 2. Whereas existing methods largely examine the boundaries of \( R_1(M) \) and \( R_2(M) \) for intersection leading to the complications described in Section 2.2, the new method uses the direct test for the intersection of the interiors of conics described in [10]. Interestingly, [10] only claims that their test is applicable to testing for the overlaps of ellipse interiors. It is conjectured by CGL that the test in [10], without modification, is also valid for testing for the overlap of the interiors of a non-singular parabola with another non-singular a parabola or an non-singular ellipse. Two of the authors of [10] indicate that this conjecture is likely to be true, and may provide a formal proof of this later if it is not already a known result (private communication). While no mathematical proof is offered here, the validity of the conjecture to the extent to which it is needed to support the functioning of the new algorithm is experimentally verified by the computation of millions of \( M_{T2} \) values and their comparison to the existing fast, slow, and analytic computations.

The benefits of this test are not only that it is one method that is as suited to working with parabolae or ellipses (or mixtures thereof) but that it also represents the conics in a matrix form whose coefficients can remain bounded even as quantities like \( m_s \to 0 \). No special change in the representations of these conics occurs when \( m_s \) or \( m_t \) move in small amounts near zero. It is these features that allows for easy machine-precision calculations of \( M_{T2} \), as no ad-hoc se-
lection criteria (cuts) are needed to deal with any of the transitions described in Sections 2.1, 2.2 and 2.3.

The second alteration is small but no less important, and concerns item 4 of section 2. Since the one place that the test of [10] can become singular is the one place where the trial has reached kinematic limit, or as close to this as the machine-precision will allow, we can use this as a stopping condition for the bisection, provided that the lower end of the initial bisection region sits on the kinematic limit. In other words, the lower end of the kinematic limit is started, by fiat, on the kinematic minimum. The upper limit is grown exponentially until it is found to be bigger than the true value of $M_{T_2}$. Thereafter the intervals are bisected, updating the upper or lower endpoints, as appropriate. If at any stage in this bisection process either of the cones look singular, it will be known this can only have happened because the trial value $M$ has been forced down to the kinematic limit, or as close to that as the machine-precision will allow. This is therefore a sign that the true value of $M_{T_2}$ either is at the kinematic limit, or is as close to that as the machine can determine. So when such a condition is encountered, the singular nature of the conic terminates further evaluation and the kinematic minimum is returned. The advantage of this is that it allows a scale-free way of dealing with the very lowest $M_{T_2}$ values.

The third and final tweak in the new algorithm is one that provides a factor of three speed increase when evaluating $M_{T_2}$ on samples in which a large fraction of their events near the lower kinematic minimum, while leading to a negligible (3%) slowdown for other events. Since any $M_{T_2}$ value returning the kinematic minimum will have produced only bisects which shrunk the bracketing interval in the high side, one can bias bisects low (say to the bottom 10% of the current bracketed range) until such time as a trial $M$ value is found to be less than the true value of $M_{T_2}$, after which bisects move to the centre of the bracketed interval. By this means, answers near the kinematic minimum can be reached roughly three times ($\sim \log_2(10)$) more quickly than they would otherwise, with negligible loss of performance for answers in the ‘bulk’. Unlike the two changes that came before, this is relatively minor as it only leads to speed increases for some classes of events.

A single C++ header file fully implementing an $M_{T_2}$ calculator using on the above method is available as part of the first arXiv submission of this paper. Links to later versions or corrections (if any) will be signposted on [6] and or in the arXiv comment field.

4 Validation

A selection of validation plots are provided here, that selectively highlight the biggest areas of difference between the new algorithm and the CH baseline. Figure 1 illustrates CH algorithm failing to accurately describe the upper end of a $t\bar{t}$ $M_{T_2}$ distribution at the 0.002 GeV level. Figure 2 illustrates the CH algorithm failing to cope with large numbers of near massless particles near the lower kinematic endpoint of another $M_{T_2}$ distribution. Figure 3 is an example of a comparison in the bulk. The comparisons in the bulk of the distribution illustrate relative differences between the algorithms. It would be ideal if both of the algorithms could be compared not only to each other but to the true value of the minimum. This is possible when the recoil (‘upstream’) momentum of the system of inputs to $M_{T_2}$ is zero - in this case there are known analytic formulae for asymmetric $M_{T_2}$ [3]. Figure 4. Compares the new algorithm with the analytic formulae in two $t\bar{t}$ events for which one of the leptons is lost and so $\chi_1 = m_\nu, \chi_2 = m_W$, and the spread of $M_{T_2}$ is $\sim 100$ GeV. The agreement between the machine precision numerical algorithm and the analytic calculation agree to better than $10^{-12}$ GeV, or one part in $\sim 10^{14}$.

5 Summary

Thus far the new algorithm has passed all validation tests it has been given, including many not described here. It can calculate both symmetric and asymmetric $M_{T_2}$ to machine-precision (one part in $\sim 10^{14}$ on the computer on which it was tested). In cases where the new algorithm has been found to produce different results to the
de-facto standard calculator (CH), the new algorithm is found to be correct. Data in Table 1 indicates that, if asked to calculate to the same 0.002 GeV precision as the old CH algorithm, the new algorithm is about two to three times faster and is capable of evaluating $M_{T^2}$ values at $\sim$MHz rates. If asked to calculate at machine-precision, the new algorithm is at most 20%-50% slower than CH, and is still capable of 200 kHz evaluation rates. There appears, therefore, to be many good reasons for trialling it a new reference implementation.

6 Acknowledgements

CGL gratefully acknowledges the assistance of Gary Hughes and Mohcine Chraibi, authors of [11], for having pointed him towards [10] after discussions about their paper, which would otherwise have formed the basis of this algorithm. CGL also gratefully thanks Joel Walker and Ben Brunt for useful discussions, and Alan Barr for making many comments on the final draft. Finally, CGL notes that three weeks after writing this algorithm, but about three weeks before writing up, he was contacted by colleague Colin Lally who, by chance, had independently come up with very much related ideas (sharing the matrix addition of conics discussed in [10] as a starting point, for example). CGL thanks Colin for suggesting ways in which the method described herein could potentially be improved. None of these ideas are yet implemented.

References

[1] C. G. Lester and D. J. Summers, Measuring masses of semi invisibly decaying particles pair produced at hadron colliders, Phys. Lett. B 463 (1999) 99–103, [hep-ph/9906349].
[2] A. J. Barr, B. Gripaios, and C. G. Lester, Transverse masses and kinematic constraints: from the boundary to the crease, JHEP 11 (2009) 096, [arXiv:0908.3779].
[3] P. Konar, K. Kong, K. T. Matchev, and M. Park, Dark Matter Particle Spectroscopy at the LHC: Generalizing $M(T^2)$ to Asymmetric Event Topologies, JHEP 1004 (2010) 086, [arXiv:0911.4126].
[4] Y. Bai, H.-C. Cheng, J. Gallicchio, and J. Gu, Stop the Top Background of the Stop Search, JHEP 1207 (2012) 110, [arXiv:1203.4813].
[5] ATLAS Collaboration Collaboration, G. Aad et. al., Search for top squark pair production in final states with one isolated lepton, jets, and missing transverse momentum in $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector, arXiv:1407.0583.
[6] A. J. Barr and C. G. Lester, “Oxbridge transverse mass library.” http://www.hep.phy.cam.ac.uk/~lester/mt2/index.html.
[7] H.-C. Cheng and Z. Han, Minimal kinematic constraints and $M_{T^2}$, JHEP 12 (2008) 063, [arXiv:0810.5178].
[8] H.-C. Cheng and Z. Han, “UCD transverse mass library.”
http://particle.physics.ucdavis.edu/hefti/projects/doku.php?id=wimpmass.
[9] J. W. Walker, A complete solution classification and unified algorithmic treatment for the one- and two-step asymmetric S-transverse mass $\tilde{M}_{T^2}$ event scale statistic, JHEP 1408 (2014) 155, [arXiv:1311.6219].
[10] F. Etayo, L. Gonzalez-Vega, and N. del Rio, A new approach to characterizing the relative position of two ellipses depending on one parameter, Comput. Aided Geom. Des. 23 (May, 2006) 324–350.
[11] G. Hughes and M. Chraibi, Calculating ellipse overlap areas, Computing and Visualization in Science 15 (2012), no. 5 291–301.
Figure 1. This figure shows a zoom in to the upper kinematic endpoint, ±0.01 GeV, of a $t\bar{t}$ $M_{T^2}$ distribution that ought to stop at 175 GeV (i.e. at zero on the $x$-axes above). It is noted that the new algorithm (left) stops at 0, as it should, while CH (right) is sometimes about 0.002 GeV too high.

Figure 2. This figure shows the lower end of an $M_{T^2}$ distribution containing a high number of close-to-massless visible and invisible objects. The kinematic minimum has been subtracted, so the distributions shown should never access negative positions on the horizontal axis. The new algorithm (left) performs correctly, but the old CH algorithm (right) shows a large delta-function spike at an unphysical (negative) position, indicating that these $M_{T^2}$ values are lower than the kinematic minimum. An unphysical mass-gap is also observed in the CH case.
Figure 3. This shows the difference between the value of $M_{T^2}$ calculated using the CH algorithm, and the value of $M_{T^2}$ calculated (for the same event) by our new algorithm, for a large number of $t\bar{t}$ events. When supplying momenta to the algorithms, the correct $l+b$ pair for each ‘side’ of each event was used. There is a narrow core of events that both algorithms get right, and a tail in which (in this case) CH happens to be high at the 0.002 GeV level. Some individual events from this tail were examined in detail (by inspecting the minimisation surfaces in arbitrary precision plotting programs and finding the location of the true minimum by hand) and the new answer was, in all cases, found to be the correct one.

Figure 4. Left: A comparison between the new algorithm and the analytic calculation in the balanced case. Right: the same comparison in the unbalanced case - now the analytic formulae has comparable calculation complexity as the numerical procedure and so the difference is symmetric about zero.
Table 1. This figure shows the time in seconds taken to perform 400,000 $M_{T2}$ evaluations using three different methods, each on two sorts of event (one for which CH tended to over-estimate $M_{T2}$ values, ‘OVER’, and one for which CH tended to under-estimate $M_{T2}$ values ‘UNDER’). All programs were compiled with -O3 optimisations turned on, and were run on a 1.4 GHz Intel Core i5 MacBook air. The three methods compared are (i) the old CH algorithm (known to be accurate to about 0.002 GeV), (ii) the new algorithm ‘LESTER’ running in machine precision mode, and (iii) the new algorithm ‘LESTER’ running to a lesser precision of 0.002 GeV. It may be seen that the new algorithm is no worse than 20% to 50% slower than CH when working in machine-precision mode (i.e. generating much better answers than CH). It is evident that when asked to only attempt to get answers to 0.002 GeV, the new algorithm was then about two to three times faster than the old CH algorithm at similar precision.