魔術師スペルの

パラメータ依存について

第一部 **INT** と **DAM** の関係について　初版 **2006/05/18**

改訂二版 **2006/06/12**

データ収集：ユーロスター

解析・文章：らりおす

調査に協力してくれた方々：黒太子，花欄，武蔵丸煮込み，緋霙，Telecaster，魔女るりCHAN，うっすぃー，ローニャ，小太朗，SilverFang，Nadeshiko，wisだいすき，どらこ，エルズ（敬称略）

概要

・ なにが書いてあるか？

int と dam どっちが強いか、数式にしてみました。

・ なにがわかるの？

今の装備で、int あげようか dam あげようか迷ってたらどっちがお徳かわかる。

装備買う前に今の装備とどっちがどのぐらい強いかとかわかる。

・ なんか面白いことわかった？

3 の倍数+1 の int 値(1,4,7,10,…)は損ってことがわかった。また、dam は奇数値(1,3,5,…)が損であるが、微々たるものである。

int1>dam1 の関係が逆転することは特例を除いてはない。でも、int が高くなるほど int1=dam1 に近づいていく。

目次

1、説明

　　魔のスペルのダメージは dam・int・MC・LV・スペルの種類に依存すると考えられる。
ここでは、ダメージは dam と int の二変数関数であるとする。

＊その他の変数については、次回以降のレポートにて検証を行う

　　int を i、dam を d、ダメージを F(i,d)とする。
　　以後のデータは、かかし MC-30　ウィンドバイン・付与あり　で算出した。
　　検証は LV95 魔、素 int98（女王討伐ボーナス 1 含む）で行った。

*例えば、int=100, dam=50 のダメージが 4528 ならば　　　F(100,50)=4528 と書く。

全部のデータをグラフにすると、以下のようになる

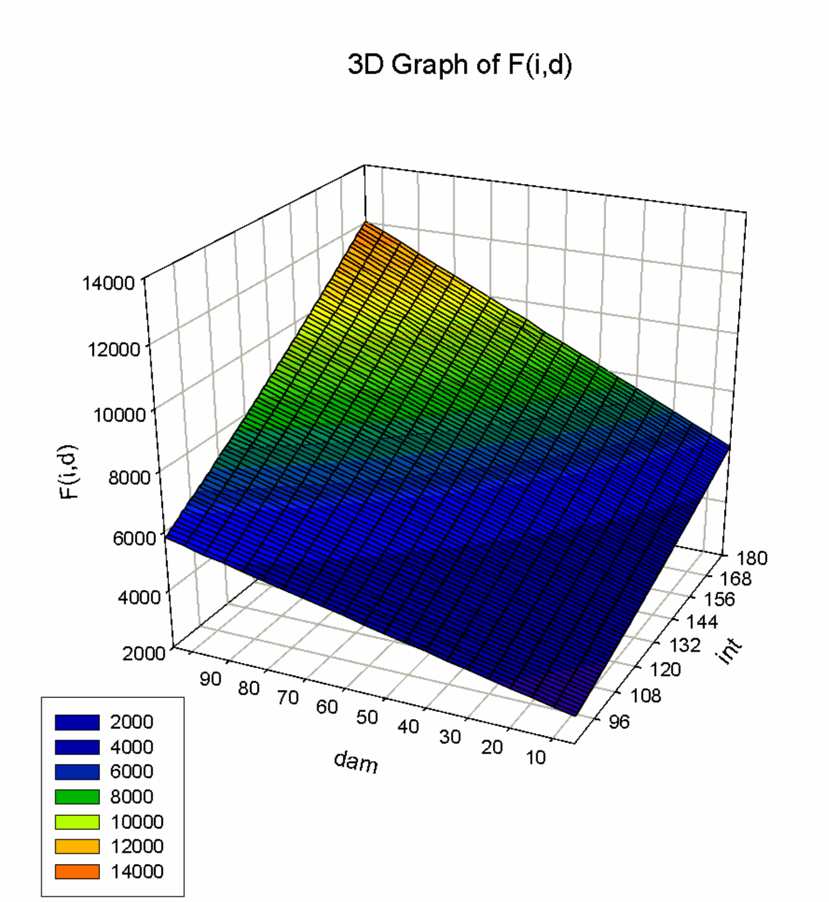


図 1:生データの F(i,d) 2 変数グラフ

int を 96~176 まで 1int 毎に、dam を 0~100 まで 5dam 毎に変化させて、与えたダメージをプロットしたグラフ

２、誤差

　　　　検証を始める前に、一定 int,dam での測定したダメージのばらつきについて調べる必要がある。ばらつきが大きいほど、誤差がでてしまい正確な式が導き出せないことになる。そこでまず、F(i=120,d=0)でのダメージを１００回測定してみた。結果以下のようなばらつきが表れた。

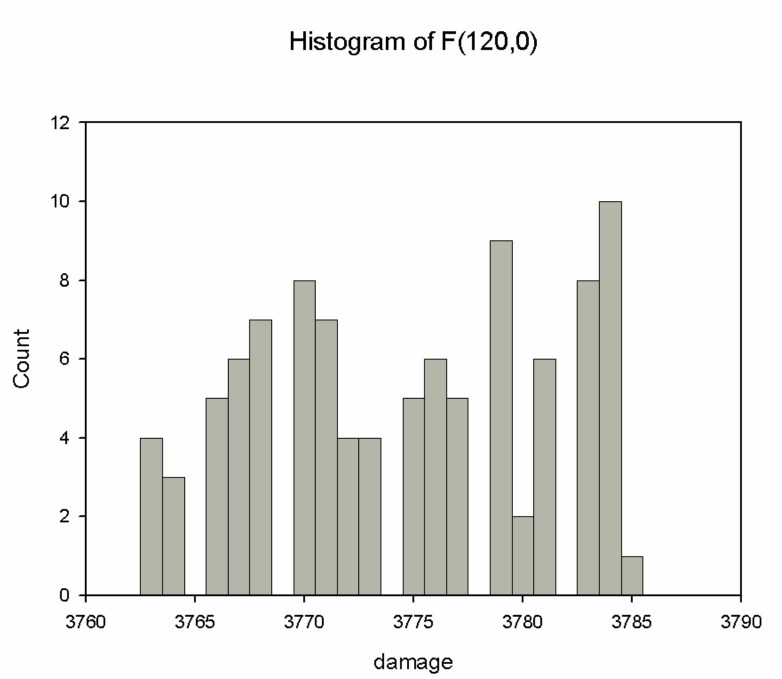


図 2: F(120,0)でのヒストグラム

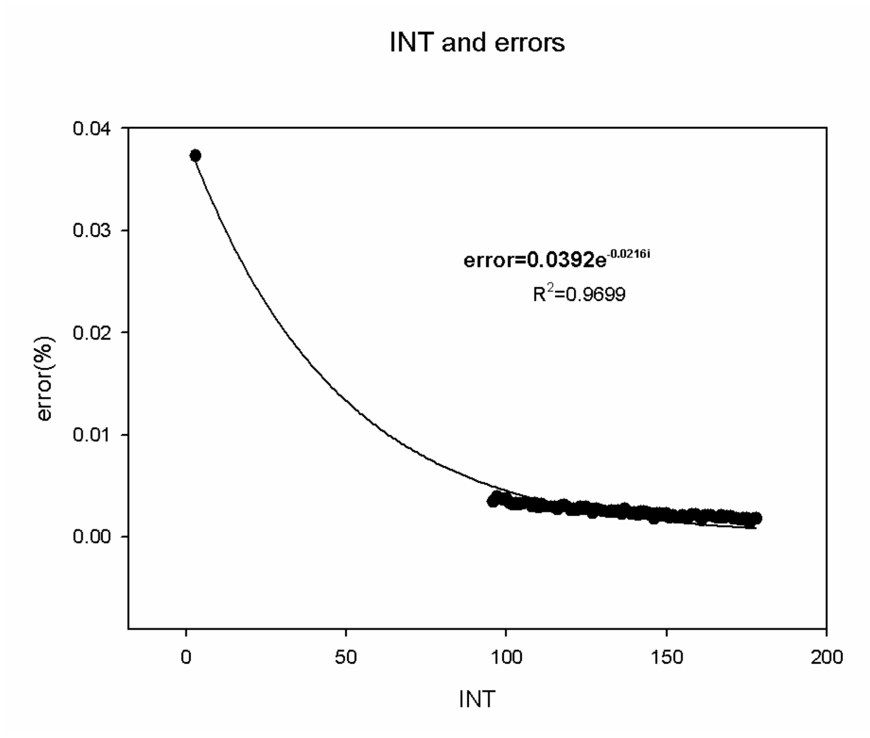
int120,dam0 の条件で 100 回ダメージを測定して、そのダメージのばらつきをみたもの。

　　　　上記のヒストグラムから、ダメージの誤差は、正規分布などではなく、完全にランダムであることが確認された。

次に、この誤差はどのような法則に従っているかを調べた。調べた条件は、F(i=96~178,d=0)を調べたもので、各々の項目につき２０回ずつダメージを測定した。

この結果、誤差は　　　　F(96,0)では 0.0034%であったのに対し、
　　　　　　　　　　　　F(178,0)では 0.0018%まで減少した。

また、別レベルのキャラによるデータとなるが、F(3,0)での誤差は 0.0373%であった。この結果をグラフにすると以下のように明らかに誤差率が減少していた。なお、近似は指数減少関数でおこなったがあまりうまく近似できなかった。試行回数が２０回なので誤差の近似は少し困難であった可能性がある。



*標準偏差:ばらつき度合い。平均値から遠い値が多いほど大きくなる。学校のテストで偏差値が高いって言うのとほぼ同義

図 3: int とダメージの誤差

int を 96~176 まで 1int 毎に変化させ、それぞれの int で 20 回ずつダメージを測定し、20 回のダメージの標準偏差を縦軸にとったもの。int3 の値は 10 回試行の参考値

そこで、２０回試行の F(i,0)の最大値と最小値の差をみてみた。すると、差はどの int 値においても、17 から 23 程度であった。例として、

$F(3,0)_{min}=219$, $F(3,0)_{max}=236$　差＝18

$F(178,0)_{min}=6151$, $F(178,0)_{max}=6173$　差＝22

であった。これらのことから、誤差は比率でなくどのような int 値においても固定であることが判る。あくまで仮説だが、このデータの場合、最大値と最小値の差はどの int 値においても 23 で、これは F(3,0)の 10%が誤差になっているのではないであろうか。

３、dam

dam は単純に割合のダメージ増加と考えられている。すなわち、

$F(i,d)=F(i,0)\left(\dfrac{d+100}{100}\right)$という式が予想される。

*つまり dam=0 でのダメージに dam%分のダメージを上乗せした値が実際のダメージ

そこで、int を固定して、dam を増減させたグラフを書いてみる。以下のグラフは int96-170 間を１つずつの間隔であげていった個々のダメージをプロットしたものである。

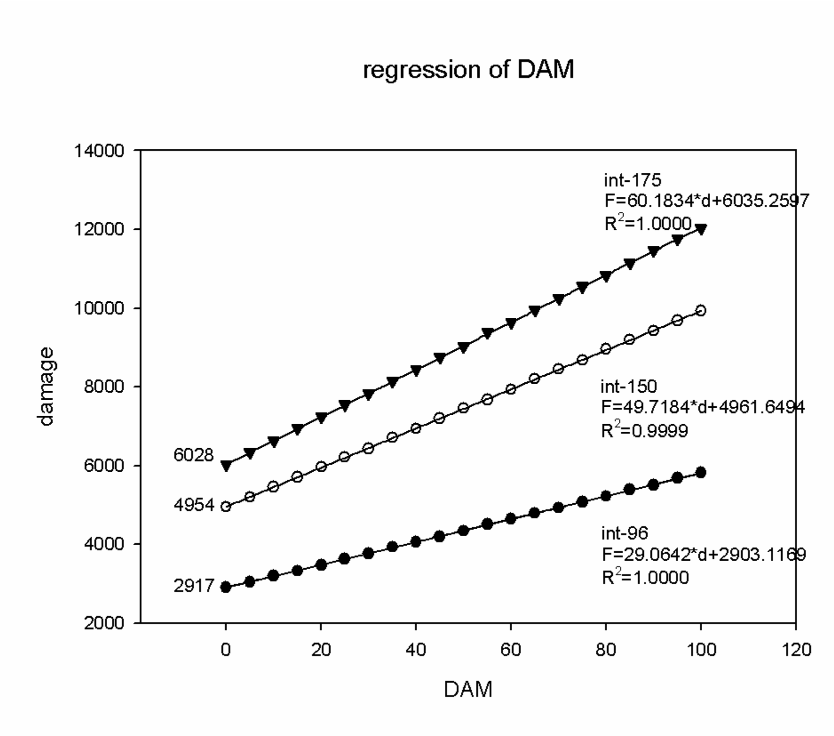


図 4: dam とダメージの相関

int を 96,150,175 で固定し、dam を 0 から 100 まで 5 毎に変化させ、グラフにしたもの。線形近似を行い線で結んだ。グラフ左の値は dam0 での値で、線形近似したときの切片とほぼ同一である

中にある数式は線形近似　(y=ax+b の形に近似したもの)を行った場合の数式である。

F(96,d)=29.06d+2903.12

F(150,d)=49.72d+4961.65

F(175,d)=60.18d+6035.26

*R^2：決定係数。この値が 1 に近いほど、その近似に沿っている

で近似できる。

これらは R^2　1.0 であり、信頼できるといえる。また逆に、二次方程式 $y=ax^2+bx+c$ に近似を行うと、R^2 は 0.85 前後となり、確からしさが落ちる。

ここで、傾き×100　切片となっているのが判る。F(96,d)ならば 29.06*100　2903.12

これらの結果より、前述の仮定は正しかったことになる。すなわち

$F(i,d)=F(i,0)\left(\frac{d+100}{100}\right)$　　であることが示された。

4、int

int の上昇はどのような関数で変化するのであろうか。公式 HP には二次関数で増加すると書いてあるが、本当にそうなのか。今回は dam を固定して int を変化させ、どのような変異があるかをグラフにした。まず、ひとつの系列につき２０回の試行を行った F(96-178,0)のエラーバー付きのグラフを書いた。

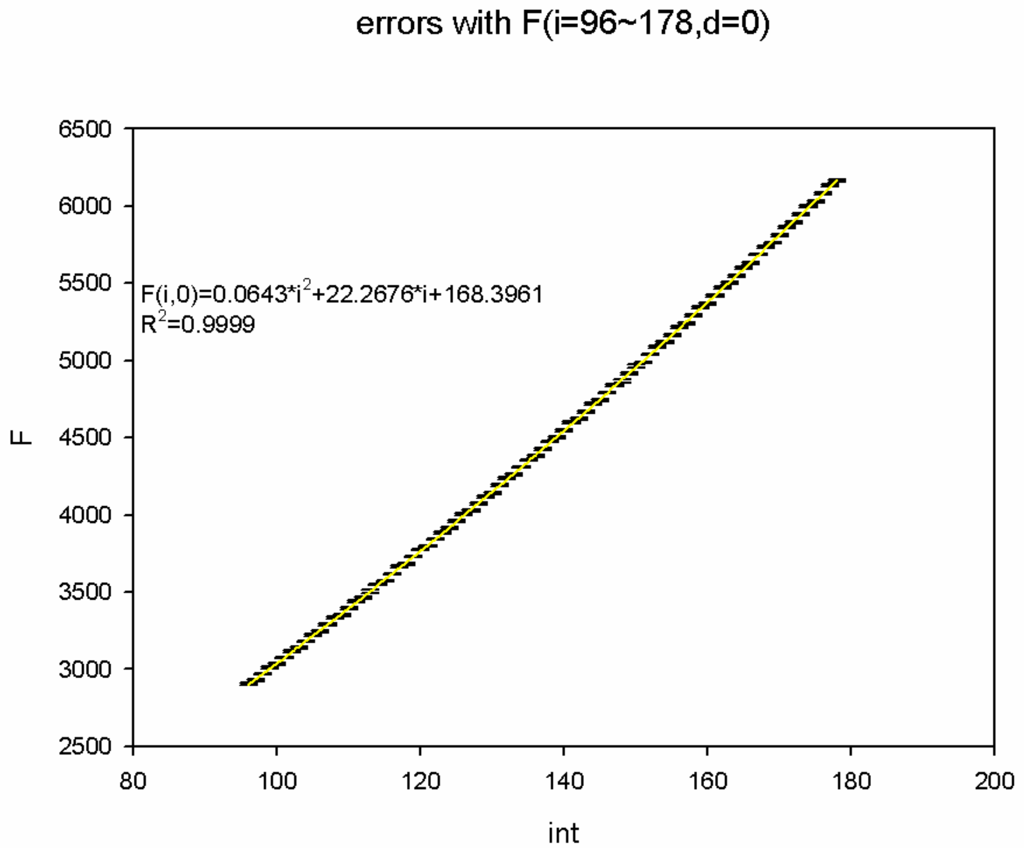


図 5: int とダメージの相関

dam0 において、int を 96 から 178 まで 1 ずつ増やしていったときのダメージを示すグラフ。各々20 回ずつのダメージを平均・標準偏差をとり、エラーバーをつけた。二次関数として近似した曲線を黄色で表す。

このグラフをみると、誤差は小さく無視できる範囲にあるといってよい。

そして F(i,0)は二次関数 $F(i,0)=0.0643i^2+22.2676i+168.3961, R^2=0.9999$ に沿った。ところで、線形近似を行うと、$F(i,0)=39.8935i-1002.6472, R^2=0.9988$ と近似の精度が下がる。また、F(3,0)=-1000 と負の値がでてしまう。これらを比較したグラフを以下に示す。

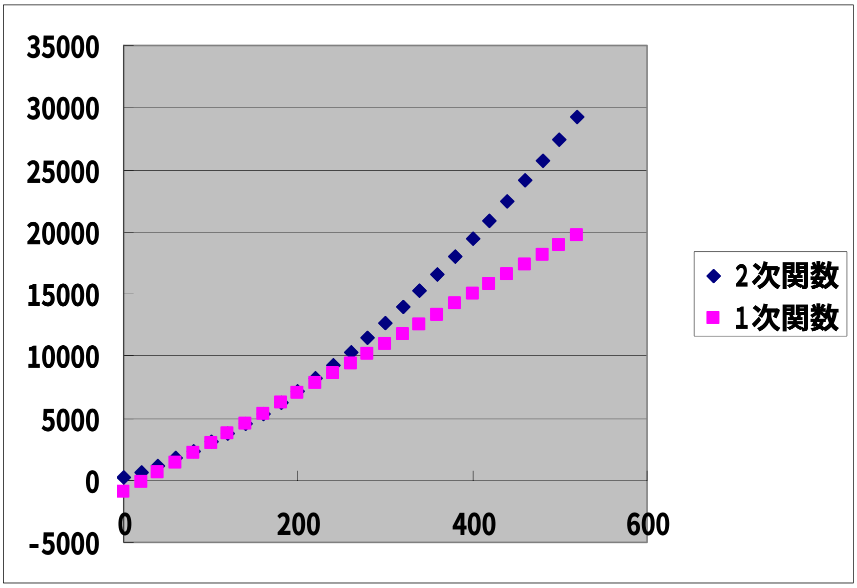


図 6: int は線形か二次関数か

図 5 での近似式をもとに、線形近似と二次関数近似のどちらが確からしいかを見たグラフ。int100-200 程度ではどちらもほとんど変わらないが、int が低い領域で線形近似だと負の値が出てしまう。

さらにここに F(i=3,0)=227 を考慮すると、二次関数での近似式に沿うことがわかる。
また、dam を変化させても下図のとおり二次関数に近似できていた。

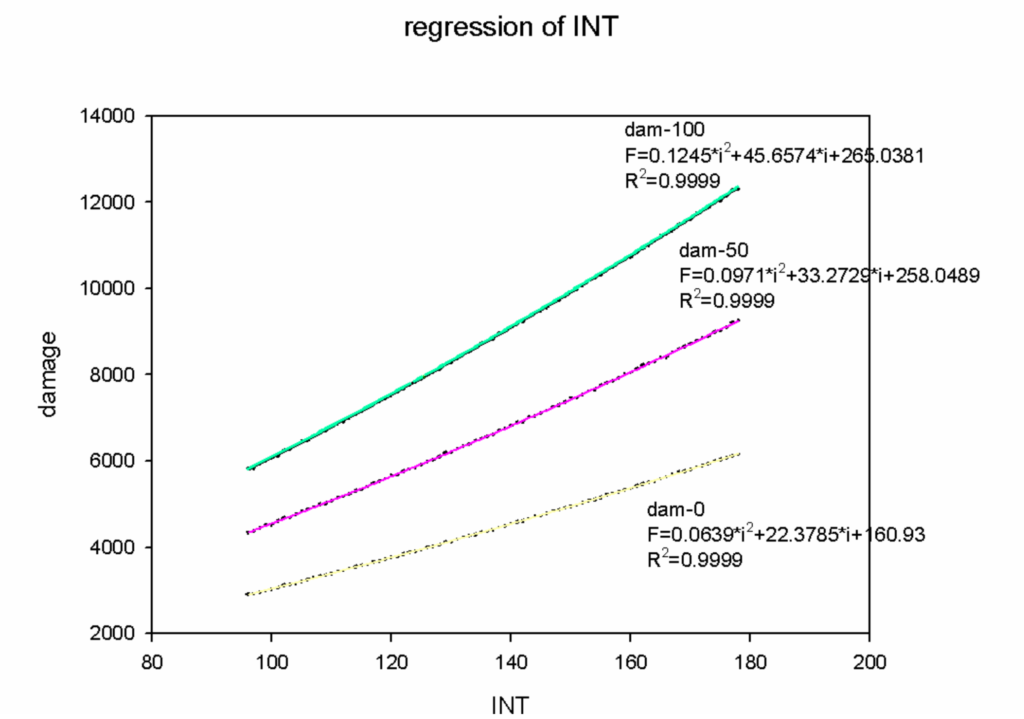


図 7: int とダメージの相関

dam を 0,50,100 で固定し、int を 96-178 まで増やしていったグラフ。二次関数でとして近似した。

以上より、int は二次関数であり

$$F(i,0)=0.0639i^2+22.3785i+160.93$$

という式に近似できた。

ちなみにこの値を二次関数の標準形で書き直すと

$$F(i,0)=0.0639(i-175.10663)^2+1799.1763$$

$$\approx 0.064(i-175)^2+1800$$

ところで、この近似は int96 以上のデータのみで行っている。低い int 域においてもこの数式は成り立つのか検証を行ったのが次の表である。

	F(3,0)	F(10,0)	F(11,0)	F(10,1)	F(11,1)
実測値	**228.6154**	**378.6333**	**408.8**	**383.6333**	**411.2667**
計算値	**228.6406**	**391.105**	**414.8254**	**395.0161**	**418.9737**

F(3,0)については１０回、その他については３０回試行の平均値を実測値として用いた。キャラクターは LV75/wis 魔で、その他のデータをとった LV95/int 魔より LV が低いため計算値よりも実測値のほうが若干ダメージが下がるが、どのダメージにおいても計算値と実測値の関係は比例しており、上記の近似式はどの int,dam 域においても成立した。なお、LV とダメージの依存については、次回のレポートを参照していただきたい。

３節・４節の結果より、

$$F(i,d)=(0.0639i^2+22.3785i+160.93)\left(\frac{d+100}{100}\right)$$

が導出された。

５、int+1＝ dam+1 の境界

導出された式は二次関数と一次関数の合成関数なので、int1 の上昇が dam1 の上昇を上回る点があると考えられる。

まず、dam+1 について考える。

$$dam+1\text{したときの上昇率}=\frac{F(i,d+1)}{F(i,d)}=\frac{(0.0639i^2+22.3785i+160.93)\left(\frac{(d+1)+100}{100}\right)}{(0.0639i^2+22.3785i+160.93)\left(\frac{d+100}{100}\right)}=\frac{d+101}{d+100}$$

同様に、int+1 では、

$$\text{int+1の上昇率} = \frac{F(i+1,d)}{F(i,d)} = \frac{(0.0639(i+1)^2 + 22.3785(i+1) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}$$

これらをグラフ化したのが以下の図である。

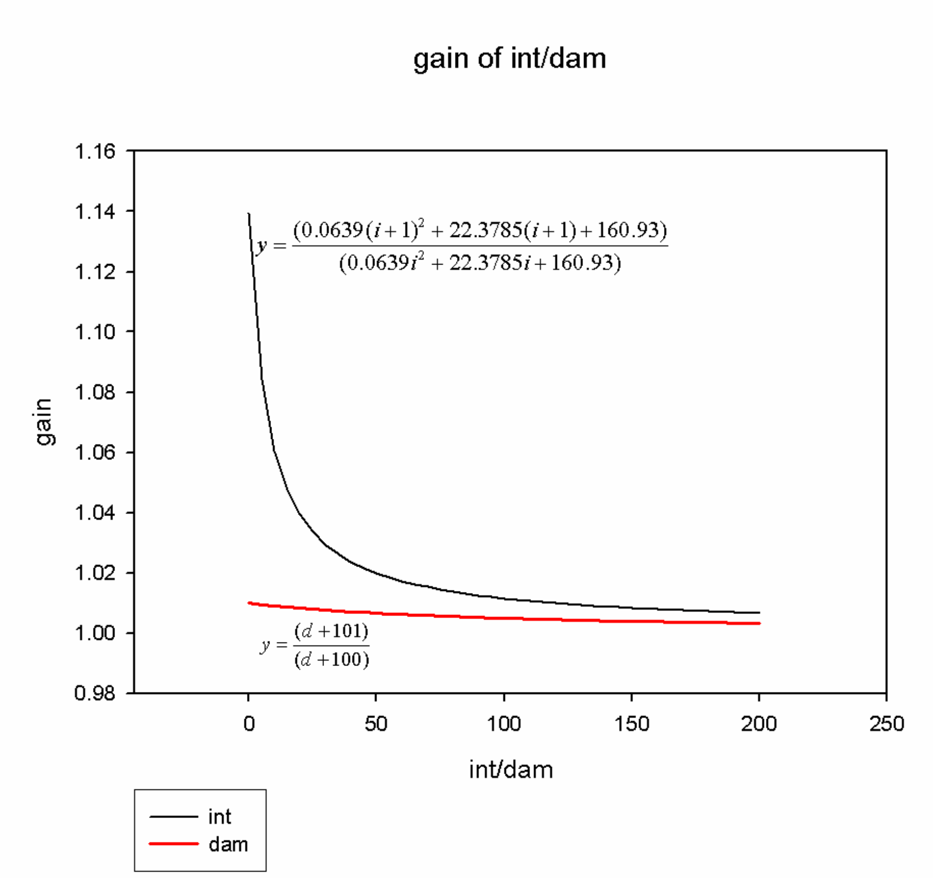


図 8: int/dam とダメージ上昇率(gain)

int と dam のダメージ上昇率を示すグラフ。gain=int(n+1)/int(n)

例えば、int100→101 にしたときの gain が 1.01 なら、int100→101 にするとダメが 1.01 倍になる。

int が低い領域(特に int100 以下)においては int1 の上昇が dam1 の上昇を大きく上回るが、高 int になるにつれ int,dam ともにダメージ上昇率が 1.00 に収束していくことがわかる。

すなわち、int200 までの間に平均の int１の上昇が dam１上昇を下回ることはない

ただし

ここで生データを振り返り、上昇率をみることにする。使用したのは２０回ずつ試行した F(i=96-178,0)である。

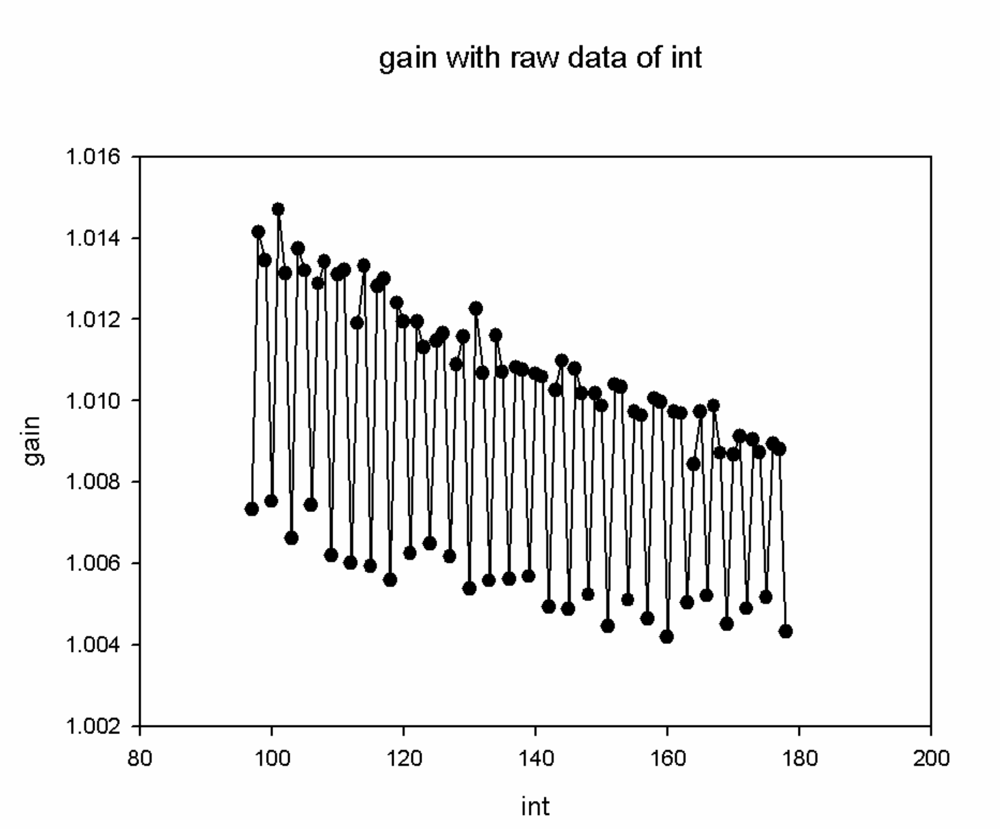


図 9: int のダメージ上昇率(gain)

int ダメージ上昇率を生データで描いたグラフ。int-1/int1 を線でつないでみると、int は 3n +1(n は整数、以下同様) の場所で小さな値をとっている。これは int 3n+1 になるときの上昇率が低いことを示す。

このグラフは、生データが実際は２つの関数から出来ていることを示す。また、3n+1(n は整数、以下同様)の int 値では低いダメージ上昇率、3n,3n+2 の int 値では高いダメージ上昇率であることがわかる。理論式では、このグラフの中間点から int の上昇率を取っていたため、この差がみえなかった。そこでこのグラフが二つの関数で出来ているとして、分けてプロットしてみる。

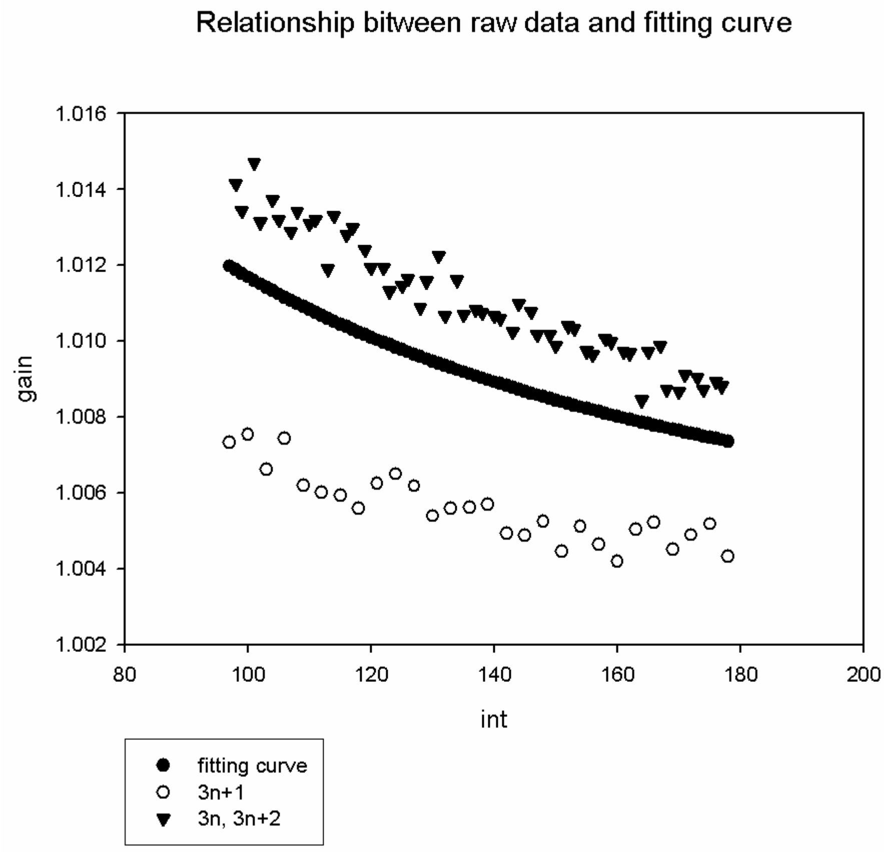


図 10: int のダメージ上昇率と近似曲線

図 9 と、図 8 の int ダメージ上昇率を合わせて書いたもの。

以上のグラフより、あくまで定性的な値だが、3n、3n+2 の int96-177 の条件では、1.001826±0.000483 倍のボーナスが、また 3n+1 の int 値には 0.99639±0.00068 倍程度の負のボーナスがあることが判った。

ところでこのような傾向は dam にもあることがわかった。dam は偶数値のとき小さなボーナスがあることがわかった。以下のグラフは、dam0 から 10 まで 1 ずつあげていき、各々20 回測定したダメージの平均値をプロットしたものである。

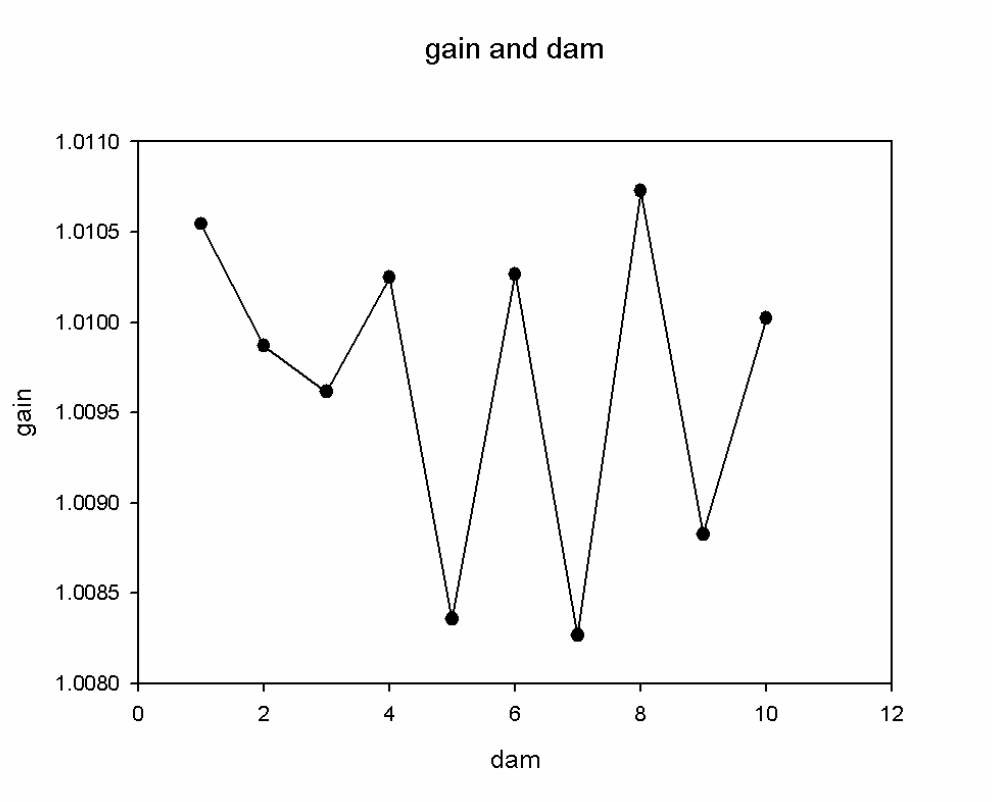


図 11: dam のダメージ上昇率

dam0 は例外であるが、dam のダメージ上昇率は、偶数 dam の時大きく、奇数 dam のとき小さい。

このように dam は偶数値(2n)において、1.00025 倍程度の補正、奇数値(2n+1)において 0.9983 倍程度の補正があることがわかった。

これらを考慮すると、int1<dam1 となる場合が出てくる。その様子をグラフにしたのが図 12 である。

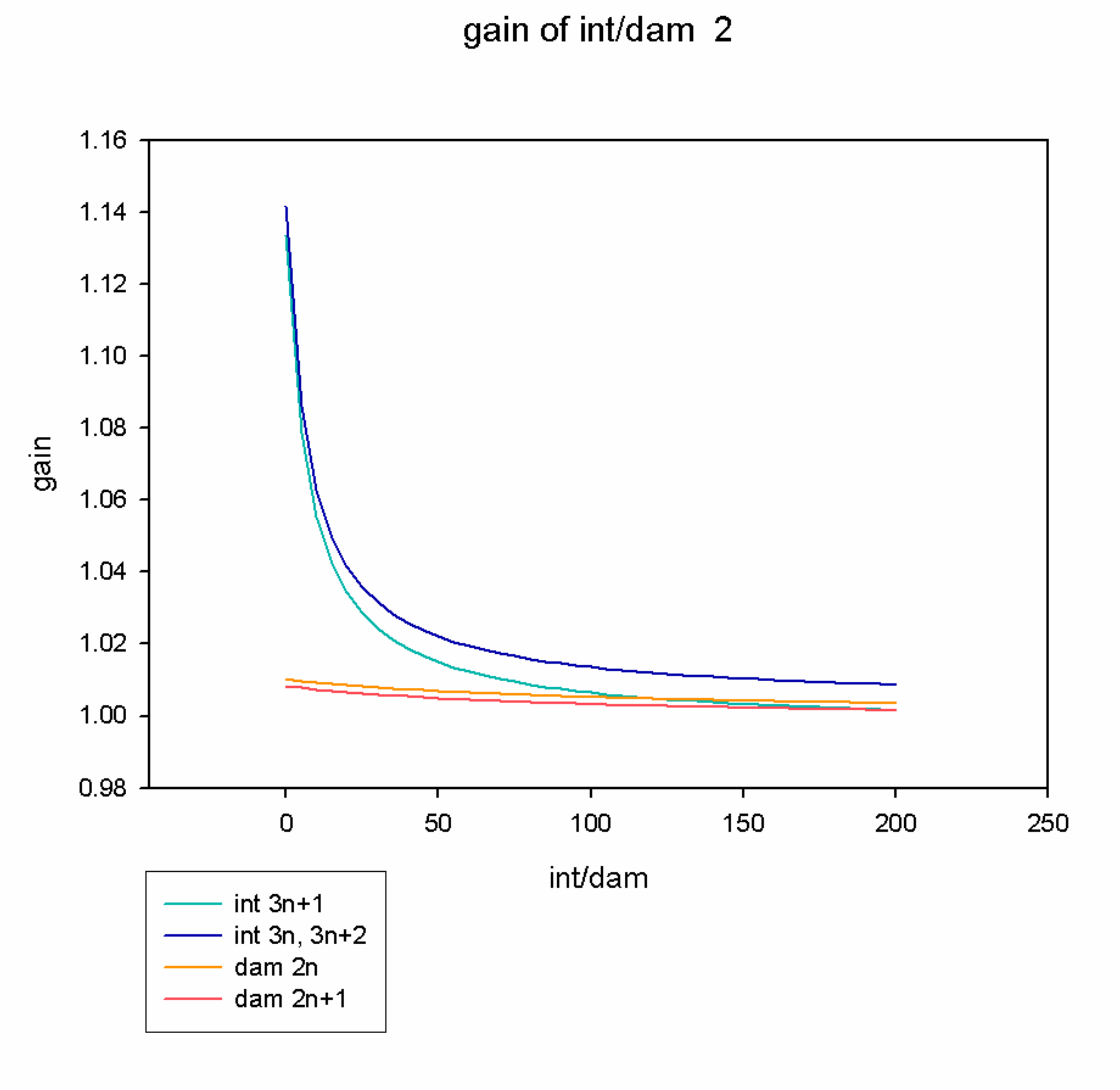


図 12: int/dam のダメージ上昇率(gain)＋3 の倍数補正

図 9 に、3 の倍数での上昇率補正を考慮したもの。int が高い領域で dam の上昇率を下回ることがわかる。

図 12 において、int の曲線が、(int/dam 軸に関係なくどこかで) dam の曲線を下回ったとき、dam1>int1 となる。

例えば、int3n+1 の曲線（水色）は、int82 で gain 1.0100 である。これは、dam2n の曲線（橙色）の dam0 での gain 1.0103 を下回っている。すなわち int82,dam0 においては dam1>int1 であることがわかる。

以上より、int1=dam1 の境界線は 4 つの場合にわけて考える必要がある。個々についてグラフを書くと、以下のようになる。

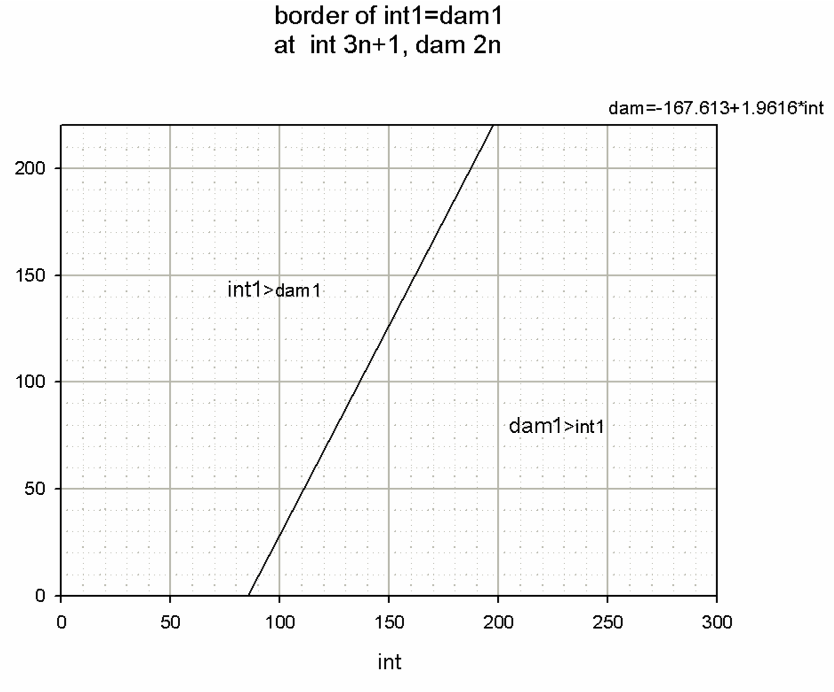


図 13: int と dam の上昇率の境界 [int3n+1 かつ dam2n の場合]

dam=-167.613+1.9616*int という数式が int1=dam1 の境界である。それより下の範囲（すなわち dam<-167.613+1.9616*int）の領域は dam1>int1 で、上の範囲は int1>dam1 である。

図 13 から、int3n+1 かつ dam 2n の時は、int82 と比較的低い int 領域で dam1>int1 となってしまう。

他の場合のグラフは、

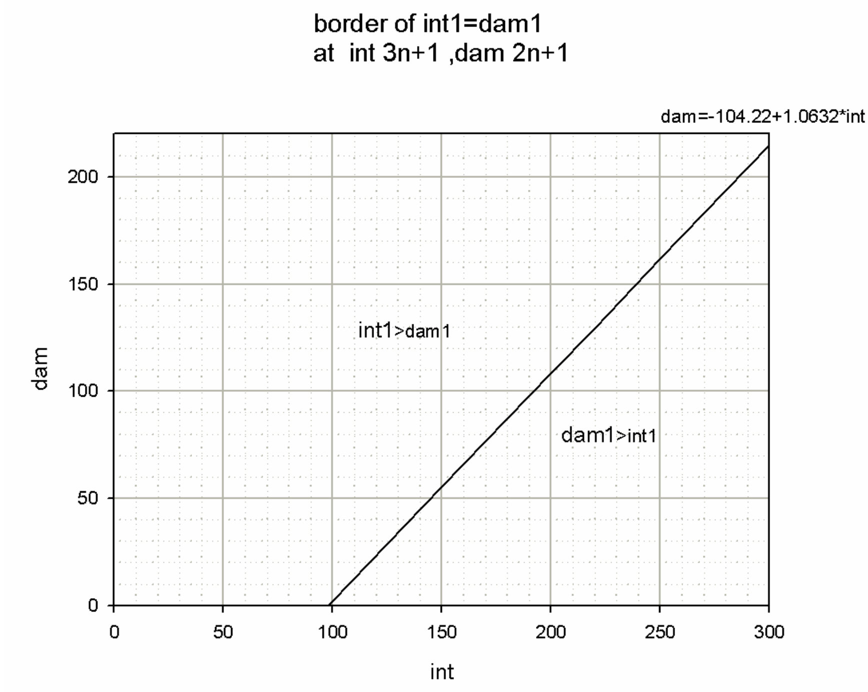


図 14: int と dam の上昇率の境界 [int3n+1 かつ dam2n+1 の場合]

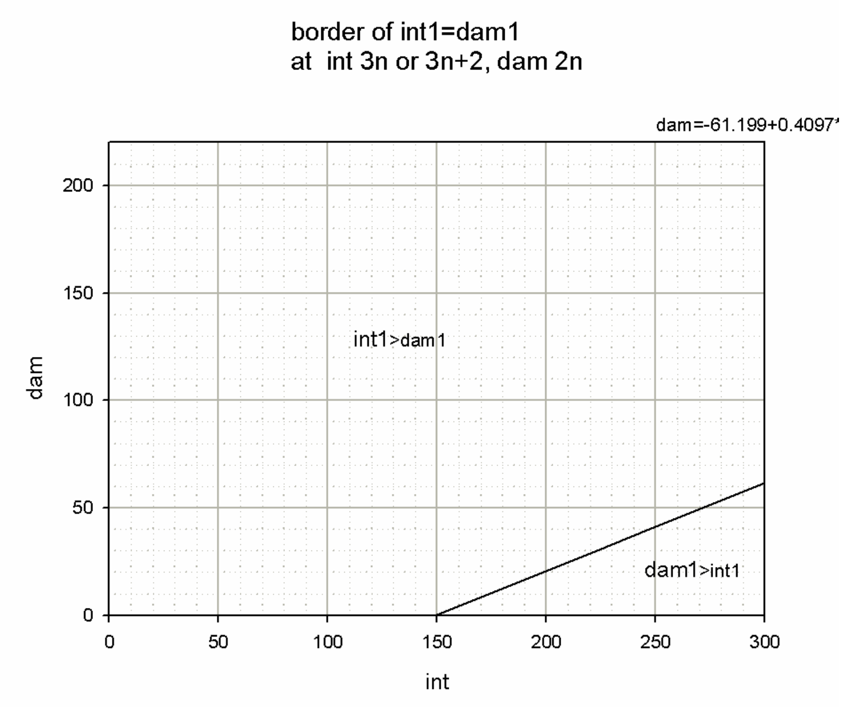


図 15: int と dam の上昇率の境界 [int3n,3n+2 かつ dam2n の場合]

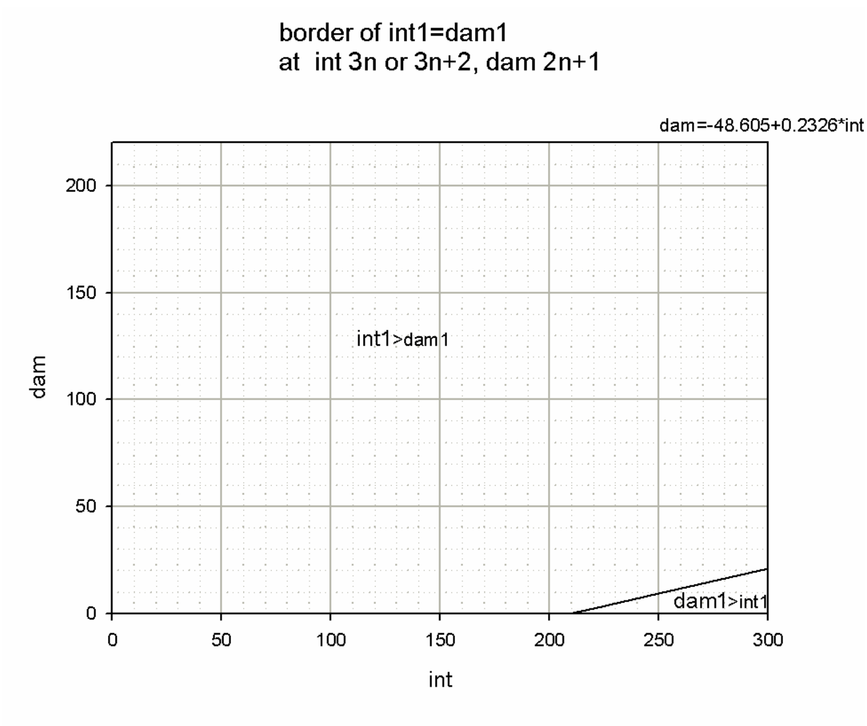


図 16: int と dam の上昇率の境界 [int3n,3n+2 かつ dam2n+1 の場合]

int 3n+1 では高い int 領域で dam1>int1 になる場合が多く存在する。しかし int が 3n+1 でないときは、極端な場合（int200 で dam20 以下の 2 の倍数など）を除いて int1>dam1 であることが判った。

6、(int+n_1,dam+n_2)＝ (int+m_1,dam+m_2)の境界

５節では int1=dam1 を探した。これを一般論まで底上げしてみる。

知りたい事：　int を n_1つあげて dam を n_2つあげるのと、
int を m_1つあげて dam を m_2つあげるの、どっちが強いか

$$n\text{の上昇率} = \frac{F(i+n_1, d+n_2)}{F(i,d)} = \frac{(0.0639(i+n_1)^2 + 22.3785(i+n_1) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+n_2+100}{100}\right)$$

$$m\text{の上昇率} = \frac{F(i+m_1, d+m_2)}{F(i,d)} = \frac{(0.0639(i+m_1)^2 + 22.3785(i+m_1) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+m_2+100}{100}\right)$$

として、n の上昇率＝m の上昇率となるところが、ボーダーライン
すなわち、

$$\frac{(0.0639(i+n_1)^2 + 22.3785(i+n_1) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+n_2+100}{100}\right) \text{、} \frac{(0.0639(i+m_1)^2 + 22.3785(i+m_1) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+m_2+100}{100}\right)$$

ゆえに

$$(0.0639(i+n_1)^2 + 22.3785(i+n_1) + 160.93)(d+n_2+100) \text{、} (0.0639(i+m_1)^2 + 22.3785(i+m_1) + 160.93)(d+m_2+100)$$

そして、自分の int を i に代入、dam を d に代入する。

int 値が 3n+1 かどうかでの判断を入れる場合､以下の作業を行う。ただし､この値は int96-177 の間の目安なので、参考値として使用する。(今後の課題でもある)

自分の int の値と目標とする int の値によって最終値に補正値を掛ける。

1）	自分の int が 3n+1、目標値が 3n+1	x0.9964
2）	自分の int が 3n+1、目標値が 3n or 3n+2	補正必要なし
3）	自分の int が 3n or 3n+2、目標値が 3n+1	補正必要なし
4）	自分の int が 3n or 3n+2、目標値が 3n or 3i+2	x1.001826

dam 値も同様に、

5）	自分の dam が 2n、目標値が 2n	x1.00025
6）	自分の dam が 2n+1、目標値が 2n	補正必要なし
7）	自分の dam が 2n、目標値が 2n+1	補正必要なし
8）	自分の dam が 2n+1、目標値が 2n+1	x0.9983

最終値は数万～数百万程度の値になるが、あくまで指標なのでこの値に意味はない。算出された二つの値を比較して、大きい値がより有利な補正であることを示す。

Example: int=100,dam=50 の人が(int+1,dam+2)するのと (int+0,dam+7)どっちが強い？

$$n\text{の上昇率} = \frac{(0.0639(i+1)^2 + 22.3785(i+1\ \ ) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+2+100}{100}\right)$$

$$m\text{の上昇率} = \frac{(0.0639(i)^2 + 22.3785(i) + 160.93)}{(0.0639i^2 + 22.3785i + 160.93)}\left(\frac{d+7+100}{100}\right)$$

よって

$(0.0639(i+1)^2 + 22.3785(i+1) + 160.93)(d+102)$、$(0.0639(i)^2 + 22.3785(i) + 160.93)(d+107)$

$i = 100, d = 50$を代入して、

$(0.0639(101)^2 + 22.3785(101) + 160.93)(152)$、$(0.0639(100)^2 + 22.3785(100) + 160.93)(157)$

今回は前者に dam 補正の必要があるので、前者に 1.00025 をかける。

また後者には int の負補正があるので、同様に後者に 0.9964 をかけると、

$467171.678 < 475172.269$

となり、後者のほうが強いことがわかる。また比をとることで、与えるダメージの差がわかる。

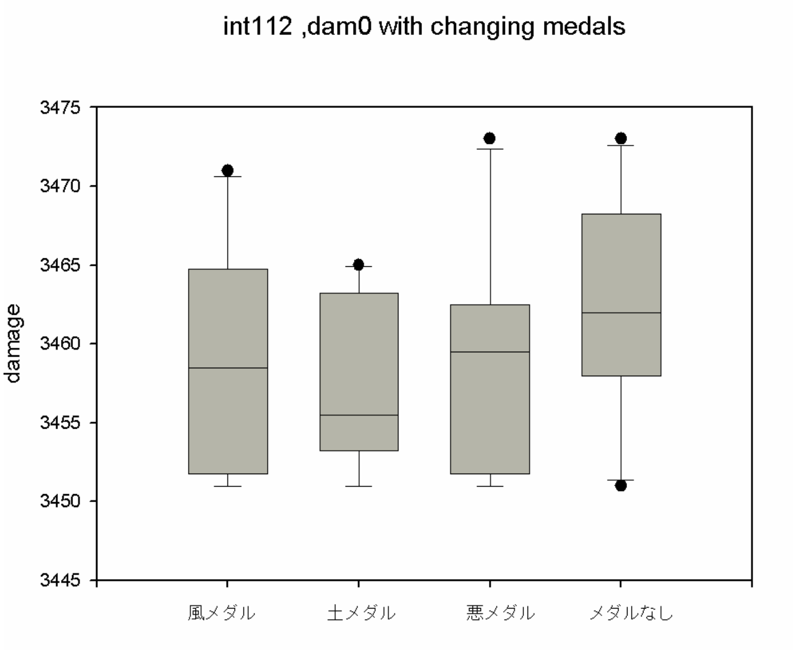
$$\frac{475172.569}{467171.678} * 100 = 101.712$$

よって後者のほうが約 2%高いダメージが期待できる。

付録１ dam0 時のメダルによる誤差

　　　メダルなしの dam0、属性合わせた dam0、悪メダルの dam0 は同じなのであろうか。

以下のグラフは、かかし MC-30　ウィンドバイン・付与あり（Lv95 素 98・INT112/DAM0 固定）の条件で、風メダル（すなわち属性合わせたメダル）、土メダル（弱点メダル）、悪メダル、メダルなしでのダメージの差を比べたものである



補図１　dam0 でのメダルとダメージの関係

線の一番上が最大値、下が最小値、箱が標準偏差、箱の中のラインが平均値である。それぞれについて１０回ずつ計測した

以上のとおり、最小値が同じ点や、標準偏差がほぼ同一である事から、メダルを変えても dam に変化がないことがわかる。つまり、dam0 の場合はメダルとスペルの属性を合わせてもダメージが変わらないことが示された。

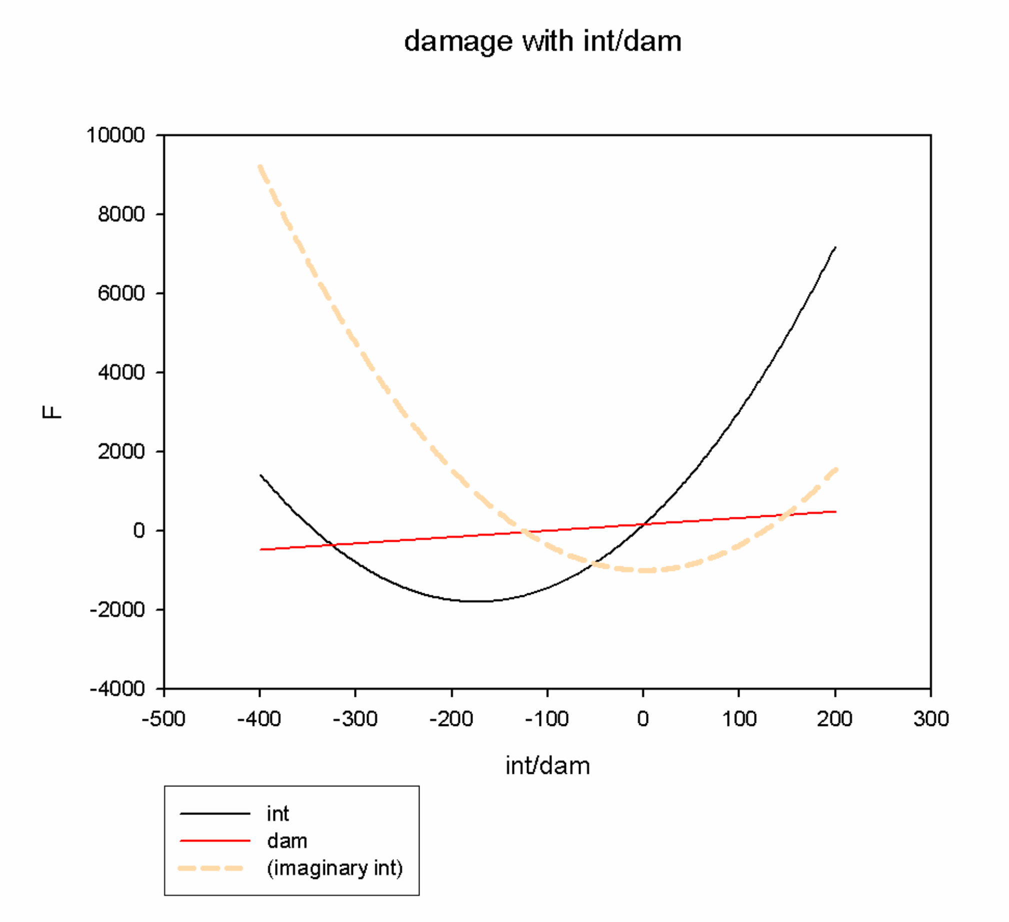
付録２　dam 式と int 式の交点

２つの関数

$$F(i,0) = 0.0639i^2 + 22.3785i + 160.93$$

$$F(0,d) = 160.93\left(\frac{d+100}{100}\right)$$

の交点を求めると



補図２　dam の関数と int の関数の交点

図のように、黒い int の関数と赤い dam の関数が 0 と-325 で交わっていることがわかる。このことから int>1 の範囲で同 int,dam のときに dam が int を逆転することは無いといえる。肌色の破線は仮定の２次曲線で、「int は２次関数だから低い int 域において dam を下回る可能性がある」という考え方がもし成り立つならば、このようになってなければならない。このように、２次関数とはいえ交点が 1~200 の間にある関数でないと上記の考えどおりにならない。