The Fast Theater Model (FATHM)

Gerald G. Brown
Alan R. Washburn
Operations Research Department
Naval Postgraduate School
Monterey, CA 93943-5000

24 January 2002

The Fast Theater Model (FATHM) is an aggregated joint theater combat model that fuses Air Force Air-to-Ground attack sortie optimization with Ground-to-Ground deterministic Lanchester fire-exchange battles using attrition rates derived from the Army's COSAGE model. The modeled FATHM war is conducted in short periods, with Air-to-Ground and Ground-to-Ground actions carried out in parallel. Period-by-period, damaged and destroyed targets may regenerate, and there may be scheduled reinforcements of attacking platforms, munitions, and new targets. The war progresses in phases whose completion depends on threshold levels of target kills in class categories and limits on phase duration. Each phase has a separate COSAGE input file, so the phases may differ strongly from each other. All FATHM inputs and outputs are ASCII flat-files suitable for immediate integration with a host database and spreadsheet analysis. Completion of a full-scale theater scenario requires about ten minutes on a personal computer.

1. Executive summary

FATHM is an aggregated joint theater combat model that quickly answers "what if" questions about the numbers and effectiveness of resources committed to battle, particularly attack platforms and munitions. Battle is viewed as consisting of two parts conducted in parallel over a period of weeks or months: the Ground-to-Ground part and the Air-to-Ground part. Sea battle is not represented, nor is air-to-air battle except by assuming that Blue controls the air throughout. Ground-to-air battle is represented only in that air strikes in the Air-to-Ground battle do carry the implication of possible attrition. Battle occurs in phases, with phase transitions depending on battle results as well as minimum and maximum phase durations.

The Ground-to-Ground part of FATHM (hereafter simply the Ground model) is a Lanchester system incorporating both direct and indirect fire, with direct fire being reprogrammed to other targets in the event that some direct target is exhausted. The required attrition coefficients are obtained by pre-processing COSAGE killer-victim scoreboards that are appropriate to the phase. As in COSAGE, Red weapons are represented explicitly.

The Air-to-Ground part (hereafter simply the Air model) consists of a sequence of sorties by Blue platforms against Red targets. Most platforms are fixed-wing aircraft, but launchers of expensive munitions such as TLAM and ATACMS are also put in this category. This part of FATHM is optimized period-by-period using Linear Programming, very much in the spirit of models such as HEAVY ATTACK and CFAM. Platforms are attrited, but there is no explicit reference to the Red assets that cause the attrition. Attrition rates can therefore depend on time, but not on battle results. The sole influence of Red fixed-wing aircraft in FATHM is to cause diversion of Blue sorties against Red targets such as airfields, and possibly to influence the time-dependent attrition rates that are determined exogenously. Because Blue air superiority is assumed, only platforms involved in direct attack are modeled (SEAD, CAP, and ECM, in particular, are not modeled).

The Air model keeps track of munitions expenditure, and will respect any munitions constraints that are imposed. Indeed, one of the purposes of FATHM is to measure sensitivity to such constraints. The FATHM objective function flexibly acknowledges the importance of

- Ending the current phase quickly;
- Assuring an equitable distribution of effort over the services;
- Avoiding attrition; and
- Killing the Red targets that are killing the most Blue platforms on the ground.

COSAGE runs typically include both Blue and Red fixed-wing aircraft, but FATHM ignores these. This is because Blue aircraft are represented instead in the Air model, and Red aircraft are assumed to cause no attrition on account of the underlying assumption of

Blue air superiority. Red and Blue helicopters, however, are included in the ground model. The ground model does not keep track of munitions expenditure, and shares no munition with the Air model.

The two models run myopically and almost independently during each time period, with an Air battle following each Ground battle. The two models influence each other . through the effect on shared Red weapon systems. Also, the lethality of each Red shooter during the latest ground battle is communicated to the Air model, conveying the contemporary value of Red shooters on the ground as targets for Air attacks. In this way the Air assets are automatically motivated to attack the Red targets that are currently being most destructive in the ground model.

The period length has so far been taken to be three days, but a principle of FATHM is that all small lengths of time should produce essentially the same results. "Essentially the same" means that smaller lengths should be more accurate but more time consuming, as in solving ordinary differential equations.

New platforms, munitions, and targets can be scheduled to arrive in-theater by time period. In addition, a destroyed Red target may be repaired and regenerated. The likelihood that such a target can regenerate depends upon whether the kill was by an air attack or from ground fire; e.g., the chances of regenerating after a ground kill can be set to zero.

All combatants are regarded as being part of the same aggregate in FATHM. The only spatial representation is the location of the Forward Edge of the Battle Area (FEBA), which moves back and forth depending on the force ratio. However, the list of targets available to the Air model may include targets not represented in the ground model, and it is in principle possible for entities to change identity in a Markov fashion as time goes by. Thus the ground model might represent only tanks while the air model represents both tanks and deep tanks, with a certain fraction of deep tanks becoming tanks at the end of each period. In this limited sense movement between regions, or at least movement between populations that inhabit regions, is possible. The Markov method is also used to model dead targets that come back to life at the end of each period, a kind of repair process.

The details of the two parts are made explicit in the following sections.

2. FATHM Lexicon for Air and Ground Warfare

FATHM fuses two well-accepted families of theater models, a ground warfare model assessing effectiveness of direct fire and indirect fire exchanges between Blue and Red ground platforms, and a linear programming model of Blue air attacks on Red ground targets (e.g., CTS, Brown, Coulter and Washburn [1994], Yost [1996]), and COSAGE (e.g., Jones [1995]). FATHM adopts terminology from both these seminal applications, but must make unambiguous distinctions when necessary for clarity when all is combined in a single model federation.

2.1. Common Terminology between Air War and Ground War

Red platform	Both target and threat in the Ground model, target in the Air model
Kill	platform is rendered harmless for the present

2.2. Air War Terminology

Aircraft	Blue platform, usually an aircraft
Weapon	air-delivered bomb or missile
Target	Red platform
Profile	outline of attack
Loadout	number of Weapons delivered in attack
Weather	flying visibility, six discrete states ranging from awful to perfect
Attrition	loss of an Aircraft by any cause
Mission	candidate method of attack, consisting of (Aircraft, Weapon, Target, Profile, Loadout, Weather, expected Attrition, and expected probability of a Kill)

2.3. Ground War Terminology

Platform	ground platform, either Blue or Red
Munition	what a platform shoots
Shooter	a platform endowed with a specific Munition for shooting
Direct Fire	Lanchester’s square law with reprogramming
Indirect Fire	Lanchester’s linear law

Figure 1 illustrates FATHM in these terms.

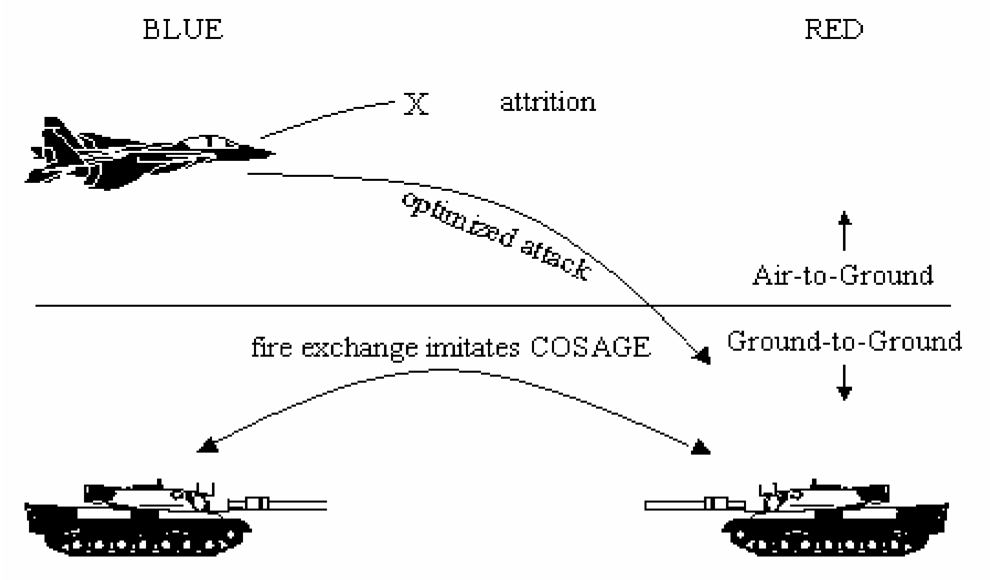

Figure 1. FATHM is a fusion of ground and air warfare. Ground-to-Ground shooting includes both direct and indirect fire, as in COSAGE. Air-to-Ground attacks are one-sided and optimized subject to constraints. The war proceeds in phases, with phase transitions depending on the number of Red platforms killed so far.

3. The Ground Model

COSAGE runs are typically for short periods of time such as two days, whereas FATHM battles may last for weeks. Although it is unlikely that any COSAGE platform will be entirely wiped out during a COSAGE run, that eventuality could very well happen in FATHM. An additional contributor to this possibility is that the Air model may concentrate on only a few target types in any one period. FATHM must therefore be prepared for the possibility that all targets of a particular type will be exhausted.

FATHM imitates COSAGE in including both direct and indirect fire. Indirect fire is not influenced by exhaustion of a target type, since the effects of indirect fire vanish naturally when the number of targets becomes zero. Direct fire, however, must be reprogrammed when no targets remain. The data structures inferred from the killer-victim scoreboards must therefore differ by type of fire.

3.1. Index use

u	Blue platform
v	Blue munition
u,v	Blue shooter
c	Red platform
d	Red munition
c,d	Red shooter

3.2. Direct fire

B_u = initial number of Blue platforms type *u*.

R_c = initial number of Red platforms type *c*.

K_{uvc} = number of kills per day by Blue *u* shooting munition *v* at Red platform type *c*.

S_{uvc} = number of shots per day by Blue *u* shooting munition *v* Red platform type *c*.

$PKdir_{uvc} = K_{uvc} / S_{uvc} > 0$ = kill probability of each shot $S_{uvc} > 0$.

$drate_{uvc} = S_{uvc} / B_u$ = rate at which each *u,v* shoots at type *c*, assuming *c* is present.

$f_{uvc} = drate_{uvc} / \sum_c drate_{uvc}$ = fraction of Blue fire at *c* using munition *v*.

All of the above are either taken directly from a killer-victim scoreboard or are ratios of such data. At time *t* later when there are B_{ut} Blue and R_{ct} Red platforms remaining, the Ground model takes the rate at which Blue *u,v kills* Red type *c* directly to be

$$direct_{uvct} = PKdir_{uvc} B_{ut} drate_{uvc} / \sum_{c:R_{ct}>0} f_{uvc} .$$

This amounts to assuming that, if any candidate type *c* gets wiped out, then the Blue fire of type (*u,v*) that was directed against *c* in the COSAGE run will be proportionally directed against other targets in the later event. If no targets have been wiped out, the sum in the denominator will be 1. Direct fire will not be "wasted" unless the sum in the denominator is 0, in which case $direct_{uvct}$ is taken to be zero for all *c*.

3.3. Indirect fire

B_u = initial number of Blue platforms type *u* (assumed positive).

R_c = initial number of Red platforms type *c* (assumed positive).

K_{uvc} = number of indirect kills of type *c* per day by Blue *u* shooting munition *v*.

$irate_{uvc} = K_{uvc} / (B_u R_c)$ = rate at which each Blue *u,v* kills Red type *c*.

The above quantities are computed from the COSAGE killer-victim scoreboard. At time *t* later when there are B_{ut} Blue and R_{ct} Red platforms remaining, the Ground model takes the rate at which Blue type *u,v* kills Red type *c* indirectly to be

$indirect_{uvct} = irate_{uvct} B_{ut} R_{ct}$.

3.4. Attrition rate polishing

If the attrition rates were actually calculated according to the above, and if those rates were then substituted into Lanchester's equations, the resulting casualties would not be as originally read from the COSAGE killer-victim scoreboard. The reason for this is that the COSAGE battle is over a significant time interval (call it T, usually two days), whereas the Lanchester period is in theory infinitisimal (in practice 3.6 hours in FATHM, since there are 10 mini-battles fought in three days). The COSAGE board contains attrition over T, but dividing that by T produces only an average or "rough" attrition rate. The COSAGE board does not contain a complete record of survivors as a function of time, which would seem to be required in order to estimate the true attrition rate.

Since the FATHM ground model is supposed to be a Lanchester model that produces the same killer-victim scoreboard as COSAGE, this potential lack of agreement is unsatisfactory. However, the rough coefficients can be easily "polished" to make the two models agree exactly. The polishing method can be most easily described using a model that is notationally simpler than the actual ground model. Suppose it were known that the differential equation $dx/dt = \alpha x(t)$ holds for *t*>0, with the initial value *x*(0) and the final value *x*(T) known, but α unknown. A rough value of α is Δ/(x(0)T), where Δ=x(T)-x(0) is the change in x over the time interval [0, T]. This is the sort of estimate described above, with Δ being the value read from the COSAGE board. The exact value of α is $\Delta / (\int_0^T x(t)dt)$, but the time record of x is not available. If the time record were available, we could calculate α, or if α were available, we could calculate the time record. The polishing procedure is an iterative procedure based on that observation:

1. Initially estimate α as $\Delta/(x(0)T)$.
2. Use the current α to solve for $x(t)$ over the interval $[0,T]$, and let $\delta = x(T) - x(0)$
3. If δ and Δ are sufficiently close, stop
4. Let $\alpha = \delta / (\int_0^T x(t)dt)$, and go back to step 2.

Upon exit from the procedure, α will be whatever parameter makes the change in x over the interval $[0, T]$ be the known value Δ. The procedure works just as well when x is multidimensional, and is the procedure used to polish the initial estimates described in paragraphs 3.2 and 3.3. In practice step 2 is always executed five times, since that seems to be sufficient to ensure convergence.

3.5. Frontal width adjustments

Obviously FATHM's results will be most accurate when FATHM's ground battles are most similar to the COSAGE battle from which the Lanchester coefficients are extracted. Nonetheless, FATHM is prepared for certain deviations. The fact that FATHM's initial platform numbers need not necessarily equal those of the COSAGE battle has already been alluded to. The flexibility in permitting this is valuable, but there is also a danger in doing so.

Suppose, for example, that initial platform numbers in the FATHM battle are all double those of the COSAGE battle. Attrition due to direct fire will double as expected, but attrition due to indirect fire (mainly artillery in most COSAGE battles) will quadruple because indirect attrition is proportional to *both* the number of shooters and the number of targets. This would be appropriate if the increased numbers of participants were still enclosed in the same COSAGE battle space, since the density of targets per unit area would be doubled and indirect fire is basically an attack on area, rather than individuals. However, the larger FATHM battle will typically be intended to occur within a larger battle space that will diffuse the targets and thereby reduce the effectiveness of indirect fire. For this reason, the FATHM user must also provide an additional parameter FEBAWID that is intended to represent the frontal width of the battle space in kilometers. If FEBAWID is larger than COSAGEWID, the frontal width of the COSAGE battle, then the indirect fire coefficients are reduced accordingly. The COSAGEWID parameter is included in the first line of the COSAGE killer-victim scoreboard.

To be precise, all indirect fire coefficients for both sides are multiplied by the ratio (COSAGEWID/FEBAWID) before being employed in FATHM's Lanchester battles. Direct fire coefficients are not adjusted. Thus, if the FATHM battle is twice as large as the COSAGE battle in all respects (frontal width as well as platform counts), then direct and indirect attrition will also be doubled.

Only the polished and adjusted coefficients $irate_{uvc}$ and $drate_{uvc}$ are subsequently employed by FATHM. As a result, if FATHM is employed with

- no Air war,
- initial platform numbers that agree with the COSAGE numbers, and
- FEBAWID=COSAGEWID

then the attrition will agree with the COSAGE killer-victim scoreboard.

3.6. Total Attrition

In total, the rate at which each Blue platform of type *u* firing munitions of type *v* kills Red targets of type *c* at time *t* is just the sum of direct and indirect terms. The indirect term is a Lanchester linear law term, and, except for the reprogramming feature, the direct term is a Lanchester square law term. There is nothing to prohibit direct and indirect attrition from the same source if the phenomenon occurs in COSAGE.

The total rate at which Red targets of type *c* disappear at time *t* is thus

$$killrate_{ct} = \sum_{uv} (direct_{uvct} + indirect_{uvct}),$$

except that the number of Red platforms of type *c* cannot be negative.

The most complicated part of these evaluations is determining $direct_{uvct}$, which involves a sum over a complicated set. Even so, the arithmetic described above can be done very quickly for hundreds of platforms and munitions. This approach resembles the ATCAL model employed by CAA, and in fact FATHM reads the killer-victim scoreboards using FORTRAN code recycled from ATCAL with the help of MAJ Jim McMullin and Mr. John Warren [2000].

With one exception, Red platforms c shooting munition d kill Blue platforms u in an exactly symmetric way. The exception is related to direct Red antiaircraft fire, and it is needed on account of the assumption of global Blue air superiority in FATHM.

The platforms involved in the Ground model and their numbers are read from a file that is separate from the COSAGE board, even though it is the COSAGE numbers that are involved in calculating the crucial coefficients *irate* and *drate*. If the COSAGE board involves platforms that are not in the Ground battle, the effect is as if they were present in zero number. The implied assumption about direct fire is that any fire directed at (say) platform RMIG25 by platform UFLAK using munition MUNI in the COSAGE battle will be redirected against other targets in the ground battle. For example, suppose UFLAK devotes half of its MUNI direct fire to RHELO and half to RMIG25 in the COSAGE battle, but that only RHELO is actually present in the ground battle. Then, according to the formula for $drate_{UFLAK,MUNI,RHELO}$, the (UFLAK,MUNI) firing rate against RHELO will be effectively doubled (divided by .5) in the Ground battle compared to what is was initially in COSAGE. This is appropriate for Red air targets because it corresponds to the assumption that Blue control of the air makes it possible for platforms like UFLAK to reprogram fire that would otherwise be directed at Red aircraft. However, a symmetric treatment would not be appropriate for Blue air targets. Red fire against Blue aircraft should not be reprogrammed in the ground model because this ground fire is the source of the assumed attrition to Blue air in the Air model. In spite of the absence of Blue aircraft in the ground model, the fire that COSAGE directs against them should be retained in FATHM. FATHM accomplishes this by essentially retaining a phantom target for certain Red antiaircraft systems. The phantom target can cause no damage to Red, but neither can Red shoot it down. For each Red shooter, at all times the phantom target attracts whatever fraction of the shooter's direct fire was allocated to Blue aircraft in the COSAGE battle. The user must provide FATHM a "nonentity list" of Blue platform types destined to become phantoms in the ground battle.

3.7. Static platform values and FEBA movement

In combat modeling, there is occasionally a need to aggregate the power of a force into a single number, generally by weighting all of the surviving platforms by a value and summing. In FATHM, this happens in the computation of FEBA movement, since the rate at which the FEBA moves is hypothesized to depend on the ratio of Blue force power to Red force power. The "static"platform values required to make this computation are part of the input database, and can be whatever the user feels is appropriate. The Army's

WEI/WUV scores are one possibility, or simple economic values could also be used. The scale is arbitrary, but must be the same for both Red and Blue.

In each period, the static values are combined with surviving platforms of each type to calculate the power of each side, and the Blue/Red power ratio R then determines the rate at which the FEBA moves. The function f(R) that converts the force ratio to a movement rate is a piecewise linear function whose coefficients are in the input database. It is a symmetric function in that f(R) = -f(1/R), so it suffices to determine f(R) for R≥ 1.0. FATHM uses the FEBA movement rate to update the location of the FEBA and write it to an output file. The FEBA location is not in any way involved with the ground battle.

Some combat models determine platform values endogenously by using the argument that the value of a platform is just the rate at which it kills value on the other side, almost a circular definition of value. The eigenvalue method and the ATCAL method are examples of this [Caldwell, et al, 1999]. The beauty of these methods is that they obviate the need for input target values, but there are also drawbacks. For example, the value of a truck (a common platform type in COSAGE) would be zero because trucks do not actually shoot at anything, and likewise the value of a communications center would be zero. For these reasons FATHM relies mainly on the aforementioned exogenous static values. The idea that the value of a platform should depend on the rate at which it can destroy value on the other side is employed in the Air model, where it is used in computing the dynamic (red) target values that enter the objective function. In this way the Blue Air forces are encouraged to direct attention to whatever Red platforms are currently proving most troublesome in the Ground battle.

3.8. Ground Battles in each FATHM time_period

During each FATHM time period, the Ground model is applied iteratively to a sequence of ten mini-battles, applying Euler’s method to approximate the solution of Lanchester’s differential equations [e.g., Hamming 1973, p. 382ff]. Each period is divided into ten mini-battles. The results are accumulated and expressed as total kills of Red platforms, $groundkills_c$, and of Blue platforms, $groundkills_u$.

In addition, the rates at which Red shooters kill Blue platforms is observed during the last mini-battle of the time period. The static Blue value killed by each Red shooter is accumulated for each Red shooting platform, and referred to as the dynamic (since it depends on the time period) value of that shooting platform.

In conducting the subsequent Air battle, the general idea is that the value of killing any Red platform should be its static value plus its dynamic value over some suitable time interval, since killing the Red platform will have the effect of later saving whatever Blue static value it would otherwise have killed. This idea will be made more precise in section 5, where the computation of $kvalue_k$ will be described in detail.

4. The Air model

The Air model is a large linear program where limited aircraft sorties are assigned to targets in a variety of conditions in an attempt to simultaneously kill targets, avoid attrition and achieve equity of effort among the services. This section gives the formulation.

4.1. Subscripts and Sets

Phase and time subscripts are suppressed in this section for simplicity of notation. The Air model is myopic in time, so there are no objectives or constraints that cross time boundaries. Some of the data is direct input, and other data is computed between periods based on results of the most recent Ground battle (these Interperiod updates are covered in section 5).

$s \in S$	set of services
$p \in P$	set of aircraft platforms
P_s	partition of P, $s \in S$
$m \in M$	set of air_weapon types
$k \in K$	set of air_target types
$a \in A$	set of air_attack profiles
$l \in L$	set of loadouts
$w \in W$	set of weather states
$j \in J$	set of target_classes
$k \in K_j$	subset of target types k referenced in target_class j

4.2. Data

$mxtargk_k$	upper bound on target type k kills (the number available)
$kvalue_k$	target value for each target k
$mxhours_{pw}$	upper bound on aircraft type p hours used in weather state w

$used_{kpw}$	hours required for an attack on target *k* by aircraft *p* in weather *w*
$mxwepns_m$	upper bound on weapon type *m* use
cap_p	capacity of aircraft *p* (used in service equity computations)
λ	multiplier for attrition in objective function
e_{apmklw}	expected kills per sortie of attack profile *a*, aircraft *p*, weapon *m* on target *k* with loadout *l* in weather *w*
att_{apmklw}	expected attrition per sortie as above
c_{apmklw}	weapons used per sortie as above
$\underline{jgoal}_j, \overline{jgoal}_j$	lower and upper goals for kills of target_class *j*
$\underline{jpen}_j, \overline{jpen}_j$	lower and upper penalties for violating kill goals for target_class *j*
$\underline{sgoal}_s, \overline{sgoal}_s$	lower and upper goals for capacity used by service *s*
$\underline{spen}_s, \overline{spen}_s$	lower and upper penalties for violating capacity goals for service *s*

Variables (all nonnegative)

X_{apmklw}	attacks
$TGTKILLS_k$	targets *k* killed
$HRSUSED_{pw}$	aircraft platform *p* hours used in weather state *w*
$PLTSLOST_p$	aircraft *p* lost
$WEPUSED_m$	air_weapons *m* used
$SVCCAP_s$	capacity used by service *s*
$UNDERKILLS_j, MIDKILLS_j, OVERKILLS_j$	under, slack, and over-kills of target_class *j*
$UNDERCAP_s, MIDCAP_s, OVERCAP_s$	under, slack, and over-achievement of service *s* goals

4.3. Formulation

Subject to the following constraints:

$KILLS_k$:

$$TGTKILLS_k = \sum_{apmlw} e_{apmklw} X_{apmklw} \qquad \forall k$$

$$TGTKILLS_k \le mxtargk_k \qquad \forall k$$

$PLATS_p$:

$$PLTSLOST_p = \sum_{amklw} att_{apmklw} X_{apmklw} \qquad \forall p$$

$$PLTSLOST_p \le mxplats_p \qquad \forall p$$

$WXHOURS_{pw}$:

$$HRSUSED_{pw} = \sum_{amkl} used_{kpw} X_{apmklw} \qquad \forall pw$$

$$HRSUSED_{pw} \le mxhours_{pw} \qquad \forall pw$$

$WEPNS_m$:

$$WEPUSED_m = \sum_{apklw} c_{apmklw} X_{apmklw} \qquad \forall m$$

$$WEPUSED_m \le mxwepns_m \qquad \forall m$$

$SERVICE_s$:

$$SVCCAP_s = \sum_{p \in P_s, w} cap_p HRSUSED_{pw} \qquad \forall s$$

$JGOAL_j$:

$$\sum_{k \in K_j} TGTKILLS_k + UNDERKILLS_j + MIDKILLS_j - OVERKILLS_j = \overline{jgoal}_j \qquad \forall j$$

$$MIDKILLS_j \le \overline{jgoal}_j - \underline{jgoal}_j \qquad \forall j$$

$SGOAL_s$:

$$SVCCAP_s + UNDERCAP_s + MIDCAP_s - OVERCAP_s = \overline{sgoal}_s \qquad \forall s$$

$$MIDCAP_s \le \overline{sgoal}_s - \underline{sgoal}_s \qquad \forall s$$

Minimize

$$-\sum_k kvalue_k TGTKILLS_k + \lambda \sum_p cap_p PLTSLOST_p$$

$$+\sum_{j}(\underline{jpen}_{j}UNDERKILLS_{j}+\overline{jpen}_{j}OVERKILLS_{j})$$

$$+\sum_{s}(\underline{spen}_{s}UNDERCAP_{s}+\sum_{s}\overline{spen}_{s}OVERCAP_{s})$$

5. FATHM Period-to-Period Accounting

5.1. *Unifying Subscripts and Sets*

$g \in G$	global_targets
$j \in J$	set of target_classes
$g \in G_j$	subset of global_targets *g* referenced in target_class *j*
$k \in K$	set of air_target types
$k(g)$	air_target *k* associated with global_target *g*
$g \in G_k$	subset of global_targets *g* referenced by air_target type *k*
$c \in C$	set of Red platform ground_target types
$c(g)$	ground_target *c* associated with global_target *g*
$g \in G_c$	subset of global_targets *g* referenced by ground_target type *c*
$t \in T$	time periods
$h \in H$	phases

5.2. *Data*

It is customary to regard an aircraft as a generator of a certain number of sorties per day, and to constrain the use of aircraft so as to prohibit using too many sorties. FATHM was developed using the idea that an aircraft is better regarded as a generator of flying hours per day, thereby recognizing that different sorties may consume different numbers of hours depending on the target, its proximity, the weather, etc. The data element hpd_{pt} introduced below is intended to be the number of flying hours generated by one aircraft of type *p* in period *t*, to be used in calculating $maxhours_{pw}$, the total number of hours available for aircraft of type *p* in weather of type *w* in the current period. Similarly, the number of hours of type *p* consumed in mounting an attack on a target of type *k* in weather *w* is $used_{kpw}$. to be used in restricting the number of sorties when the Air model is solved. That was the plan, and both the FATHM code and the description given below agree with it. However, the authors have been unable to find a database with $used_{kpw}$ in it. As an interim measure, hpd_{pt} can be (and has been, in practice) set to the number of sorties per day while $used_{kpw}$ is set identically to 1.0; this is simply a return to the idea that aircraft generate sorties rather than flying hours. The rest of the data introduced below is more straightforward:

$inittrg_g$	initial targets of type *g*
$gvalue_g$	static value of each platform *g*
$tcon_g$	expected regeneration time in days for target *g*
$airfsk_g$	fraction of air-killed targets *g* that can regenerate
$groundfsk_g$	fraction of ground-killed targets *g* that can regenerate
$mxplatu_u$	initial Blue ground platforms *u*
$rstart_c$	initial Red ground platforms *c*
$groundkills_c$	ground kills of Red platforms *c* during period *t*
$groundkills_u$	ground kills of Blue platforms *u* during period *t*
$rdyval_c$	the total rate per day of killing Blue static value during the last mini-battle of the Ground war by one Red platform of type *c*
tau	weight of dynamic value in total value of each air target (days)
$kvalue_k$	total value of air target *k*
$teksfirst_{apmklw}, tekslast_{apmklw}$	air_attack effectiveness at the start of the war, and after learning by experience conducting the war
$attrfirst_{apmklw}, attrlast_{apmklw}$	air_attack attrition at the start of the war, and after learning by experience conducting the war
f_w	fraction of time in weather state *w*
hpd_{pt}	hours per day for platform *p*, period *t*

len_t	length of period t in days
$inacftp_p$	initial platforms p
$initwep_m$	initial weapons m
$mxregen_g$	pool of targets g that may regenerate
$kill_{hj}$	minimum kill fraction for phase h, target set j
$tglive_g$	live targets g at the start of period t
$airkills_k$	air kills during period t
$relax_p$	expected learning time in days of combat experience for platforms p
$addtrg_{gt}$	new targets g added at end of period t
$addplp_{pt}$	new platforms p added at end of period t
$addplu_{ut}$	new platforms u added at end of period t
$addwep_{mt}$	new weapons m added at end of period t
$\underline{sfrac}_s, \overline{sfrac}_s$	minimum, maximum fraction of effort for service s

Before the first FATHM period:

$h \Leftarrow 1$

$t \Leftarrow 1$

Global targets:

$$mxtargs_g \Leftarrow inittrg_g \qquad \forall g$$

$$mxregen_g \Leftarrow 0 \qquad \forall g$$

Conditions for ground battles:

COSAGE board (posture) selected with its exchange rates

$$B_u = mxplatu_u \qquad \forall u$$

$$R_c = \sum_{g \in G_c} inittrg_g \qquad \forall c$$

Conditions for air attacks:

$$mxtargk_k = \sum_{g \in G_k} inittrg_g \qquad \forall k$$

$$mxplats_p \Leftarrow initplt_p \qquad \forall p$$

$$mxweps_m \Leftarrow initwep_m \qquad \forall m$$

$$mxhours_{pw} \Leftarrow f_w hpd_{p1} len_1 initplt_p \qquad \forall pw$$

$$used_{kpw} \equiv 1 \qquad \forall kpw$$

$$e_{apmklw} \Leftarrow teksfirst_{apmklw} \qquad \forall apmklw$$

$$att_{apmklw} \Leftarrow attrfirst_{apmklw} \qquad \forall apmklw$$

After ground battles in period t and before air battle in period *t:*

Create air war total target values by aggregating the weighted sum of static and dynamic global target values. This calculation is complicated by the varying levels of platform resolution in both the Air and Ground war. The quantity $rdyval_c$ from the ground war is the rate at which any Red platform c in the Ground war kills Blue target value. Multiplying this by the time constant τ converts it to Blue value killed, but there may be several Red global platforms that correspond to Ground Red platform *c*. Thus

$$dvalue_g = (\tau)(rdyval_c);\ g \in G_c$$

is the dynamic value of any Red platform. The total value is the sum of the static and dynamic values:

$$tvalue_g = dvalue_g + gvalue_g;\ g \in G_c$$

Finally, since air target *k* is one of a set of global targets G_k that are indistinguishable from the air, the value of one air target of type *k* is obtained by averaging over the set of indistinguishable global targets:

$$kvalue_k = \sum_{g \in G_k} (tvalue_g inittargs_g) / (\sum_{g \in G_k} inittargs_g) \qquad \forall k$$

The intermediate quantities $dvalue_g$ and $tvalue_g$ play no explicit role in the Air model, and are introduced here only for clarity.

At the end of period *t (order is important here)*:

Ground war results:

$groundkills_c$ and $groundkills_u$ from ground battles

Air war results:

$$airkills_k = \sum_{apmlw} e_{apmklw} X_{apmklw} \qquad \forall k$$

Global target status:

$$mxtargs_g \Leftarrow mxtargs_g MAX[0, \quad 1 - airkills_{k(g)} / \sum_{g' \in G_{k(g)}} mxtargs_{g'} \quad - groundkills_{c(g)} / \sum_{g' \in G_{c(g)}} mxtargs_{g'}] \qquad \forall g$$

Transition tests for time_periods, phase durations, and target_class phase goals:

Too many time_periods? $t \overset{?}{\geq} |T|$ "war is over"

Not enough phase duration? $\sum_{t' \leq t \text{ in phase } h} len_{t'} \overset{?}{<} hmnlen_h$ "continue phase h"

Target_class phase goals achieved?

$$\left(\sum_{g \in G_j} inittrg_g - \sum_{g \in G_j} mxtargs_g \right) / \sum_{g \in G_j} inittrg_g \overset{?}{\geq} kill_{hj} \qquad \forall j$$

"phase h completed"

Phase h completed? $h \overset{?}{\geq} |H|$ "war is won"

Phase duration expired? $\sum_{t' \le t \text{ in phase } h} len_{t'} \overset{?}{\ge} hmxlen_h$ "phase h terminated"

Phase h terminated? $h \overset{?}{\ge} |H|$ "war is over, but not won"

Phase transition: $h \Leftarrow h+1$ "new phase h begins"

COSAGE board (posture) selected

Global target status update:

$$mxtargs_g \Leftarrow mxtargs_g + \left(1 - e^{-len_t / tcon_g}\right) mxregen_g + addtrg_{gt} \qquad \forall g$$

$$mxregen_g \Leftarrow e^{-len_t / tcon_g} mxregen_g$$

$$+ airfsk_g airkills_{k(g)} / \sum_{g' \in G_k} mxtargs_{g'}$$

$$+ groundfsk_g groundkills_{c(g)} / \sum_{g' \in G_{c(g)}} mxtargs_{g'} \qquad \forall g$$

Ground war update:

$$B_u \Leftarrow mxplatu_u \Leftarrow mxplatu_u - groundkills_u + addplu_{ut} \qquad \forall u$$

$$R_c \Leftarrow rstart_c \Leftarrow \sum_{g \in G_c} mxtargs_g \qquad \forall c$$

Air war update:

$$mxtargk_k = \sum_{k \in G_k} mxtargs_g \qquad \forall k$$

$$mxplats_p \Leftarrow mxplats_p - \sum_{amklw} att_{apmklw} X_{apmklw} + addplp_{pt} \qquad \forall p$$

$$mxweps_m \Leftarrow mxweps_m - \sum_{apklw} c_{apmklw} X_{apmklw} + addwep_{mt} \qquad \forall m$$

$$e_{apmklw} \Leftarrow e^{-len_t / relax_p} e_{apmklw} + \left(1 - e^{-len_t / relax_p}\right) tekslast_{apmklw} \qquad \forall apmklw$$

$$att_{apmklw} \Leftarrow e^{-len_t / relax_p} att_{apmklw} + \left(1 - e^{-len_t / relax_p}\right) attrlast_{apmklw} \qquad \forall apmklw$$

$$mxhours_{pw} \Leftarrow f_w hpd_{p,t+1} len_{t+1} mxplats_p \qquad \forall pw$$

$$\underline{jgoal}_j \Leftarrow \overline{jgoal}_j \Leftarrow \sum_{g \in G_j} mxtargs_g - \sum_{g \in G_j} inittrg_g(1 - kill_{h,j}) \qquad \forall j$$

$$\underline{sgoal}_s \Leftarrow \sum_{p \in P_s, w} cap_p mxhours_{pw} \underline{sfrac}_s \qquad \forall s$$

$$\overline{sgoal}_s \Leftarrow \sum_{p \in P_s, w} cap_p mxhours_{pw} \overline{sfrac}_s \qquad \forall s$$

$$t \Leftarrow t + 1$$

6. Implementation and Output

FATHM has been implemented as a master compiles FORTRAN program *fathm.exe* that runs other programs such as *fathmi.exe* to read the numerous input files, pavgr.exe to read the COSAGE killer-victim scoreboards required for all war phases and compute and store the COSAGE coefficients, fathmx.exe to perform the period-to-period recursion using GAMS/XA, and *fathmr.exe* to clean up intermediate files at the end. Output files include *fathm.log* (a cumulative history of all actions including reading the input files, any errors or inconsistencies discovered, and the results of each of the engagements), and *Attacks.csv*. *Attacks.csv* is a comma-delimited, global attack history. Each line of this primary output file shows the results of a particular attack, whether ground or air.

With one exception, input files are intended to be manipulated within an Excel workbook *FathmInputs.xls*, which has one sheet for each input file and a macro that writes the sheets to separate files. The exception is *mission.csv*, a large, rarely edited file that includes data about all of the possible air strike missions. *FathmInputs.xls* also includes an import macro.

A second workbook *Attacks.xls* is useful for viewing and summarizing the primary output file *attacks.csv*. This workbook capitalizes on Excel features such as graphics, autofiltering, and pivot tables. A "refresh" macro in *Attacks.xls* loads the latest output from FATHM. Another useful device is to link a database to attacks.csv and use queries to recover such things as overall killer-victim scoreboards, etc.

FATHM requires about 10 minutes to run a realistically scaled problem on a one GHz WINTEL microcomputer with 128MB RAM.

7. References

Brown, G.G., Coulter, D.M., and Washburn, A.R. [1994], "Sortie Optimization and Munitions Planning," Military Operations Research, 1-1, pp. 13-18.

Caldwell, W.; Hartman, J.; Parry, S.; Washburn, A.; and Youngren, M. 2000. *Aggregated Combat Models*, lecture notes, Operations Research Department, Naval Postgraduate School, http://www.nps.navy.mil/orfacpag/notes.htm, ch. 6.

Hamming, R.W. [1973], Numerical Methods for Scientists and Engineers, McGraw-Hill.

Jones, H.W. [1995], "COSAGE User's Manual, Volume 1 – Main Report. Revision 4,"

Center for Army Analyses CAA-D-93-1-REV-4 (August) 112 pages.

Warren, J. [2000], "PAVGR, a FORTRAN subroutine to read COSAGE killer-victim scoreboards" (private communication), Center for Army Analyses, June.

Yost, K.A. [1996], "The Time Strike Munitions Optimization Model," Naval Postgraduate School NPS-OR-96-001 (January) 67 pages.

8. Appendix: Procedure and Input Data Files

8.1. Essential input files

The following input files are required:

mission.csv

FathmInputs.xls

- *Cap.csv*
- *Eset.csv*
- *Feba.csv*
- *HGoals.csv*
- *Nonentity.csv*
- *PArrivals.csv*
- *Phase.csv*
- *Platform.csv*
- *Scalars.csv*
- *SGoals.csv*
- *Target.csv*
- *TArrivals.csv*
- *Tlen.csv*
- *WArrivals.csv*
- *Weapon.csv*
- *WFrac.csv*

Also required are any COSAGE boards as listed in *phase.csv* .

8.2. Program Components

The following components constitute the FATHM federation:

Component	File	Function
FATHM	*fathm.exe*	Overall executor
PAVGR	*pavgr.exe*	COSAGE data extracts
FATHMI	*fathmi.exe*	Index and data import, edits
FATHMX	*fathmx.exe*	Fights war
OPTIMIZER	*fathm.gms*	Time_period air-attack model
SOLVER	GAMS and/or XA	Linear program solver
FATHMR	*fathmr.exe*	Cleans up debris

8.3. Sequence of computations

The controlling program *fathm.exe* should be run from DOS, and should reside in the same directory that includes all the input files and all of the other executable files.

Fathm.exe first calls *Pavgr.exe* to determine the Lanchester coefficients for the ground battle. *Pavgr.exe* reads *Phase.csv* to determine which COSAGE boards (postures) are applicable to each war phase, and *Nonentity.csv* to obtain a list of COSAGE platforms that remain targets in the ground battle even though they do no damage (these are generally Blue aircraft—see section 3.5). *Pavgr.exe* then reads each of the COSAGE boards, extracts direct and indirect firing rates, and synthesizes for later use the phase-wise Lanchester coefficients of Blue shooters firing at Red platforms in *uATr.csv*, and Red Shooters firing at Blue platforms in *rATu.csv*. These two files are intermediate products that constitute all FATHM knows about COSAGE.

Once *Pavgr.exe* finishes, *Fathm.exe* calls *Fathmi.exe* to read the rest of the input files and report any indiscrepancies. *Fathmx.exe* then fights the ground battles, sets up the linear programs that represent the air battles, calls the solver to solve them, advances time, and determines phase transitions until the war is over. *Fathmx.exe* produces a complete history of every action throughout the war in *Attacks.csv*. *Fathmr.exe* deletes certain intermediate files after *Fathmx.exe* is done with them.

8.4. Index Set Definitions

FATHM insists that all input files use entity indexes consistently, with the standard for each index being taken from the key or “grounded” file given in the following table.

Index	Entity	Grounded source	field	restrictions
a	air_attack	*Mission.csv*	5	X,X,G
p	aircraft	*Platform.csv*	1	U,N,G
u	b_platform	*Platform.csv*	2	U,N,X
s	Service	*Platform.csv*	3	X,X,G
m	air_weapon	*Weapon.csv*	1	U,X,G
v	b_shooter_munition	*uATr.csv*	3	X,N,X
g	global_target	*Target.csv*	1	U,X,G
k	air_target	*Target.csv*	2	X,N,G
c	r_platform	*Target.csv*	3	X,N,X
j	target_class	*Target.csv*	4	X,X,G
y	target_category	*Target.csv*	10	X,N,X
d	r_shooter_munition	*rATu.csv*	3	X,N,X
l	loadout	*Mission.csv*	6	X,X,G
w	weather state	*WFrac.csv*	1	U,X,G
t	time period	*TLen.csv*	1	U,X,G
h	war phase	*Phase.csv*	1	U,X,G
e	mrc	*ESet.csv*	1	U,X,G

Restrictions indicate, respectively, whether or not each index instance in its grounded source must be unique (U), whether a null (empty, or all-blank) field entry is admissible (N), and whether the index must be admissible as a GAMS name (G).

Each index must be 1-9 characters long. GAMS indices must be alphanumeric with no embedded blanks or special characters.

For example, air_weapon indices m are explicitly defined for FATHM by *Weapon.csv* along with initial inventories, so each index must be unique, non-null, and must be GAMS admissible.

For example, the loadout indices l are filtered from the *Mission.csv* file, where each index may occur many times, null entries are tolerated, and each index must be GAMS admissible.

For example, the r_platform indices need not be GAMS admissible.

8.5. *Index sets and map sources*

Red targets are distinguished by a global_target name, and an air_target type and/or a ground_target type. Thus, each global_target may be known as an air_target, or as a ground_target, or as both. For example:

Global_target *g*	**air_target *k***	**ground_target *c***
T80near	tank	rT80
T72near	tank	rT72
T72deep	tank	
TruckAAA	AAA	rTRUCK
Truck		rTRUCK

In this example, air attacks do not distinguish between types of tanks, or their proximity to the ground front(s). By contrast, ground fire is only exchanged with tanks near the front(s), and the type of tank engaged makes a difference in the outcome. Air attacks focus on truck-mounted AAA batteries, but ignore just plain trucks. Ground fire engages all trucks equally.

Fusing these different views of air and ground attackers requires the following index sets and maps:

Set	**Map**	**source**	**fields**	**Set definition**
G_k	$k(g)$	*Target.csv*	2(1)	global_targets that are air_target type *k*
G_c	$c(g)$	*Target.csv*	3(1)	global_targets that are ground_target *c*
G_j	$j(g)$	*Target.csv*	4(1)	global_targets that are in class *j*
P_s	$s(p)$	*Platform.csv*	3(1)	air_platforms of service *s*
U_s	$s(u)$	*Platform.csv*	3(2)	ground_platforms of service *s*

8.6. *Extrinsic Data Sources*

Each FATHM scenario extracts its input data from the following ASCII flat-files. Note that FATHM only filters data attributes from these sources for entities with grounded indices. The names, definitions, and sources of data attributes follow.

Data	Source	field
$teksfirst_{apmklw}, tekslast_{apmklw}$	*Mission.csv*	12,13
$attrfirst_{apmklw}, attrlast_{apmklw}$	*Mission.csv*	9,10
$inittrg_g$	*Target.csv*	5
$tcon_g$	*Target.csv*	6
$airfsk_g$	*Target.csv*	7
$groundfsk_g$	*Target.csv*	8
$gvalue_g$	*Target.csv*	9
f_w	*WFrac.csv*	2
len_t	*TLen.csv*	2
$inacftp_p$	*Platform.csv*	4
$inplatu_u$	*Platform.csv*	4
cap_p	*Platform.csv*	5
hpd_{p1}	*Platform.csv*	6
$pvalue_p$	*Platform.csv*	7
$uvalue_u$	*Platform.csv*	7
$initwep_m$	*Weapon.csv*	2
$kill_{hj}$	*HGoals.csv*	3
$addtrg_{gt}$	*TArrivals.csv*	3
$addplp_{pt}$	*PArrivals.csv*	4
$addplu_{ut}$	*PArrivals.csv*	4
hpd_{pt}	*PArrivals.csv*	5

$addwep_{mt}$	*WArrivals.csv*	3
λ	*Scalars.csv*	1
tau	*Scalars.csv*	2
FEBAWID	*Scalars.csv*	3
c_{apmklw}	*Mission.csv*	6
$\underline{sgoal}_s, \overline{sgoal}_s$	*SGoals.csv*	2,3
$\underline{spen}_s, \overline{spen}_s$	*SGoals.csv*	4,5
fields present but not used:	*Mission.csv*	7,8
	Target.csv	9