RECEIVED
FEB 17 1996
OSTI

**Probabilistic Model, Analysis and Computer Code
for Take-off and Landing Related Aircraft Crashes
into a Structure**

Ron Glaser

February 6, 1996

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

OK

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

©1996. The Regents of the University of California. All rights reserved.

This work was produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U. S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL. Copyright is reserved to the University for purposes of controlled dissemination, commercialization through formal licensing, or other disposition under terms of Contract 48; DOE policies, regulations and orders; and U.S. statutes. The rights of the Federal Government are reserved under Contract 48 subject to the restrictions agreed upon by the DOE and University as allowed under DOE Acquisition Letter 91-7.

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

**PROBABILISTIC MODEL, ANALYSIS, AND COMPUTER CODE FOR
TAKE-OFF AND LANDING RELATED AIRCRAFT CRASHES INTO A
STRUCTURE**

Ron Glaser

Engineering Research Division
Lawrence Livermore National Laboratory
University of California
Livermore, CA 94550
February 6, 1996

ABSTRACT

A methodology is presented that allows the calculation of the probability that any of a particular collection of structures will be hit by an aircraft in a take-off or landing related accident during a specified window of time with a velocity exceeding a given critical value. A probabilistic model is developed that incorporates the location of the each structure relative to airport runways in the vicinity; the size of the structure; the sizes, types, and frequency of use of commercial, military, and general aviation aircraft which take-off and land at these runways; the relative frequency of take-off and landing related accidents by aircraft type; the stochastic properties of off-runway crashes, namely impact location, impact angle, impact velocity, and the heading, deceleration, and skid distance after impact; and the stochastic properties of runway overruns and runoffs, namely the position at which the aircraft exits the runway, its exit velocity, and the heading and deceleration after exiting. Relevant probability distributions are fitted from extensive commercial, military, and general aviation accident report data bases.

The computer source code for implementation of the calculation is provided.

TABLE OF CONTENTS

	page
1. Introduction.	4
2. Aircraft accident model for crashes in the vicinity of an airport.	5
2.1. Off-runway accidents.	9
2.1.1. Obtaining an expression for $P(E \mid C_o)_s$.	10
2.1.2. Obtaining an expression for $P(E \mid C_o)_g$.	19
2.1.3. Obtaining an expression for $P(E \mid C_o)_d$.	22
2.2. Runway accidents.	22
2.3. Probability distributions and parameter settings used in applications.	27
2.3.1. Off-runway impact location distributions.	28
2.3.2. Off-runway heading distributions.	34
2.3.3. Off-runway impact velocity, impact angle, and deceleration distributions.	41
2.3.4. Runway run-off and overrun distributions and parameters.	56
3. Limitations and Caveats.	62
4. Computer Codes.	65
4.1. The Fortran code for Power Macintosh computation of $P_W(E)$.	65
4.2. The input file, "crin".	97
4.3. The input file, "cring".	100
4.4. The output file, "crout".	112

1. Introduction.

A sensitive structure, such as a storage facility that houses a nuclear device, may be located close enough to an airport that an estimate is desired of the probability that the structure would be struck by an aircraft using the airport with a velocity sufficient to cause a significant problem, e.g., a radioactive release. Because of the assumed proximity of the structure to the airport, only take-off and landing related accidents are pertinent in the sense of providing non-negligible contributions. Factors in the probability model include

- the location of the structure relative to the airport's runways
- the size of the structure
- the sizes, types, and frequency of use of aircraft which take-off and land at the airport
- the relative frequency of take-off and landing accidents by aircraft type
- properties of off-runway crashes, namely impact location, impact angle, impact velocity, and the heading and deceleration after impact
- properties of runway overruns and run-offs, namely the position at which the aircraft exits the runway, its exit velocity, and the heading and deceleration after exiting.

Extensive data bases gleaned from commercial, military, and general aviation aircraft accident reports allow the practical calculation of the probability of interest. The development is motivated by the form of data available for commercial jet aircraft. However, the military and general aviation data bases

are sufficiently different from the commercial data base that separate modeling approaches are needed in some respects. It is expected that the results offer more generality than required in many applications. For example, it might be argued that the structure is sufficiently far away from the airport that a runway overrun or run-off poses no credible threat. Likewise the threshold velocity of interest might be zero, i.e., it is of interest merely to estimate the probability that the structure would be hit at any velocity.

2. Aircraft accident model for crashes in the vicinity of an airport.

The objective is to derive a computable expression for the probability that a particular structure located in the vicinity of an airport is hit during a specified window of time by an aircraft with a velocity exceeding a given critical value. Let E denote the event that the structure is hit by an aircraft with velocity exceeding v_c , where v_c may be 0. Let $P_W(E)$ denote the probability that E occurs in the window W , where W is a time interval of specified length, e.g., one year. An expression for $P_W(E)$ is obtained by taking into account the relevant contributions by runway and by aircraft type. Since the location of the structure relative to a given runway is needed in the model, and runways are typically used in either direction, each physical runway at an airport leads to two conceptual runways, depending on the direction of intended travel (Figure 1). Because the structure is located in the vicinity of the runway, an aircraft will hit the structure only in the take-off (which includes climb) or landing (which includes landing approach) phases. [En route phase crashes into the structure which involve flights that are not scheduled to use the airport near the structure are viewed as contributing a

Figure 1. Each physical runway gives rise to two conceptual runways, depending on the direction of intended travel.

negligible quantity to $P_W(E)$.] Consequently,

$$P_W(E) = \sum_{j,i} N_{Tji} P_{Tji} + \sum_{j,i} N_{Lji} P_{Lji}, \quad (1)$$

where the summation extends over all aircraft types j and conceptual runways i , N_{Tji} denotes the number of attempted take-offs of a type j aircraft from runway i during the window W , P_{Tji} denotes the probability that E occurs given an attempted take-off of a type j aircraft from runway i , and N_{Lji} and P_{Lji} are the corresponding quantities for attempted landings. The accuracy of (1) requires the conditional probabilities P_{Tji} and P_{Lji} to be small. These

probabilities are computed by considering whether the event is initiated by an impact (i.e., crash) off (i.e., away from) the runway. Let C_o denote a crash/accident in which the initiating impact occurs off the runway, and let C_r denote anything else. C_r thus denotes a runway accident, i.e., an overrun or run-off in which the aircraft leaves the runway, and an initiating impact, if there is one, occurs on the runway. Therefore,

$$P_{Tji} = P_{Tji}^o + P_{Tji}^r \text{ and } P_{Lji} = P_{Lji}^o + P_{Lji}^r, \quad (2)$$

where the off runway and runway components are broken down conveniently as

$$P_{Tji}^o = P(C_o | T)_j P(E | T, C_o)_{ji},$$

$$P_{Tji}^r = P(C_r | T)_j P(E | T, C_r)_{ji},$$

$$P_{Lji}^o = P(C_o | L)_j P(E | L, C_o)_{ji},$$

$$\text{and } P_{Lji}^r = P(C_r | L)_j P(E | L, C_r)_{ji}. \quad (3)$$

The conditional probabilities $P(C_o | T)_j$, $P(C_r | T)_j$, $P(C_o | L)_j$, and $P(C_r | L)_j$ may be estimated by relative frequencies gleaned from accident statistics: e.g., $P(C_o | T)_j$ may be approximated by the relative frequency of attempted take-offs by a type j aircraft that result in an off-runway crash. Computation of the remaining conditional probabilities, such as $P(E | T, C_o)_{ji}$, which denotes the probability E occurs if a type j aircraft taking off from runway i crashes off the runway, are discussed below. Whenever runway i is being considered a Cartesian coordinate system in reference to this runway is used. The x -axis is coincident with the extended runway center line. By transforming appropriately the joint probability density function of impact location (X, Y) , there is no loss of generality in assuming that all intended take-offs and landings go from right to left, and the structure is situated above the runway.

See Figure 2. The runway is assumed to have length q and width $2l$. The origin is at the center of the threshold end for a landing; the point $(-q, 0)$ is at the center of the departure end for a take-off. See Figure 3.

Figure 2. Stochastically equivalent situations. There is no loss of generality in assuming the direction of intended travel is right to left and the structure is situated above the runway. The bottom depiction will be used in the analysis.

Figure 3. Coordinate system.

A computer code for the calculation of $P_W(E)$ is presented in Section 4.

2.1. Off-runway accidents.

Consider the computation of $P(E \mid T, C_o)_{ji}$ and $P(E \mid L, C_o)_{ji}$. For the sake of brevity, denote either by $P(E \mid C_o)$. Conceptually, there are three mutually exclusive and collectively exhaustive ways the structure can be hit with a velocity exceeding v_c , given an initializing impact off the runway:

- "sliding hit, s": the aircraft impacts off the runway away from the structure and slides, or skids, into the structure at a velocity exceeding v_c
- "glancing hit, g": the aircraft strikes the structure on its way down with an impact velocity exceeding v_c , although it would have missed it if the structure's height were 0

- "direct hit, d": the aircraft strikes the structure on its way down at a velocity exceeding v_c , and it would have hit it even if the structure's height were 0.

It is convenient to express $P(E \mid C_o)$ in terms of the three contributing components s , g , and d ,

$$P(E \mid C_o) = P(E \mid C_o)_s + P(E \mid C_o)_g + P(E \mid C_o)_d. \quad (4)$$

The mathematical development of expressions for the component probabilities is simplified by the approximating assumptions that the aircraft may be regarded as a disk of radius w (where w is one-half the wing span) and the structure may be regarded as a cylinder of radius r and height h . The center of the structure is assumed to be located at (x_t, y_t) , where $y_t > 0$.

2.1.1. Obtaining an expression for $P(E \mid C_o)_s$.

In order for E to occur by a slide, or skid, the following sequence of events must take place:

- The initializing impact must occur off the runway somewhere outside the circular direct hit region.
- The heading after impact must be in such a range that the aircraft can slide toward the structure.
- The length of the portion of the slide during which the velocity exceeds v_c must be sufficiently long.

Specifically,

$$P(E \mid C_o)_s = \iint_{(C \cup R)} f_{X,Y}(x,y) \int_{\theta_1(x,y)}^{\theta_2(x,y)} f_{\Theta \mid X,Y}(\theta \mid x,y) P[D_s > d_e(x,y,\theta)] d\theta dy dx, \quad (5)$$

where (x, y) is the location of the initializing impact, i.e., the center of the disk of radius w ; $f_{X,Y}(x, y)$ is the joint probability density function (pdf) of the

impact location; $f_{\Theta|x, y}(\theta|x, y)$ is the conditional pdf of the aircraft's heading Θ after impact at (x, y) ; $[\theta_1(x, y), \theta_2(x, y)]$ is the range of headings for which the aircraft given sufficient momentum would ultimately slide into the structure; D_s is the length of the portion of the slide during which the velocity exceeds v_c ; $d_e(x, y, \theta)$ is the "effective distance" from the aircraft to the structure, i.e., the distance the aircraft must slide from (x, y) at heading θ in order to hit the structure; C is the circular direct hit region, i.e., $C = \{(x, y): (x - x_t)^2 + (y - y_t)^2 \leq (r + w)^2\}$; and R is the rectangular runway region which has vertices $\{(0, l), (-q, l), (-q, -l), (0, -l)\}$.

Useful expressions for the heading range $[\theta_1, \theta_2] = [\theta_1(x, y), \theta_2(x, y)]$ are obtained below. Also convenient expressions for the probability $P[D_s > d_e(x, y, \theta)]$ are derived that involve

- the pdf of deceleration after initializing impact, the pdf of impact velocity, and the distribution of impact angle given impact velocity (commercial aircraft)
- the probability of a sufficiently small impact angle, and the pdf of deceleration and the distribution of horizontal velocity, given a sufficiently small impact angle (military aircraft)
- the pdfs of impact angle and impact velocity, and the distribution of skid distance given impact angle and velocity (general aviation aircraft).

The probability distributions that are used for heading, deceleration, skid distance, and impact location, velocity, and angle depend on whether the impact is take-off or landing related and are discussed in Sec. 2.3.

Expressions for the heading range depend on the location of the aircraft at initializing impact relative to the location of the structure. For the sake of modeling symmetry with respect to the runway center line, it is reasonable to

measure the heading angle clockwise from 9 o'clock if the initial impact is above the runway center line ($y > 0$), and counter clockwise from 9 o'clock if $y < 0$. Thus positive headings imply movement away from the runway center line, and negative headings imply movement toward the runway center line. See Figure 4.

Figure 4. Headings are measured from 9 o'clock in such a way that positive (negative) headings imply movement away from (toward) the runway center line.

Consider the situation in which the initializing impact occurs above the runway center line ($y > 0$) and also to the right of and below the structure, i. e., $x > x_t$ and $y < y_t$. Refer to Figure 5. The aircraft can slide into the structure if its heading Θ falls in the range $[\theta_1, \theta_2] = [\alpha_1, \alpha_2]$ where these limits are determined geometrically. Since $\sin(\beta + \alpha_1) = (y_t - y)/d$, where $d = \sqrt{(x - x_t)^2 + (y - y_t)^2}$ is the distance between the disk centers, and $\alpha_2 = \alpha_1 + 2\beta$, it is clear that a computable expression for β will yield α_1 and α_2 . By similar triangles, $s/r = t/w$. Since $s + t = d$, it follows that $t = d/(1 + r/w)$. Therefore $\sin \beta = w/t = (w + r)/d$. Moreover,

and

A geometric diagram illustrating a layout, likely a runway or road intersection. The diagram features a horizontal line at the bottom labeled "runway". Two circles are positioned above this line. The left circle has a radius labeled r . The right circle has a radius labeled w . A horizontal line segment connects the centers of the two circles, with an angle labeled $\beta + \alpha_1$ at the right center. A vertical line segment extends from the center of the left circle down to the horizontal line connecting the centers. A line segment labeled s connects the center of the left circle to a point on the right circle. A line segment labeled t connects the center of the right circle to the same point on the left circle. Angles α_1 and α_2 are marked at the intersection of the horizontal line connecting the centers and the horizontal line at the bottom. Angles β are marked at the centers of both circles. A line labeled α_1 extends from the intersection point towards the top right. A line labeled α_2 extends from the intersection point towards the bottom right. A line labeled α_1 also extends from the intersection point towards the top left.

For the same situation but with the impact below the runway ($y < 0$), the heading range is $[\theta_1, \theta_2] = [\alpha_2, \alpha_1]$; angles are now expressed in the counter clockwise sense from 9 o'clock so that α_1 and α_2 are -1 times their respective values in the $y > 0$ case:

$$\alpha_2 = -\sin^{-1}\left(\frac{y_t - y}{d}\right) - \sin^{-1}\left(\frac{w + r}{d}\right) \quad (8)$$

$$\alpha_1 = -\sin^{-1}\left(\frac{y_t - y}{d}\right) + \sin^{-1}\left(\frac{w + r}{d}\right) \quad (9)$$

Heading limits for all other possible orientations are obtained in an analogous fashion. When the direction of movement is right to left an additive factor of π or $-\pi$ is present. The results are summarized in Table 1.

Table 1. Heading ranges $[\theta_1, \theta_2]$ for slides into the structure.

$$\text{Definitions: } \alpha = \sin^{-1}\left(\frac{|y_t - y|}{d}\right)$$

$$\beta = \sin^{-1}\left(\frac{w + r}{d}\right)$$

Abbreviations:	+	$y > 0$ (impact above runway)
	-	$y < 0$ (impact below runway)
	R	$x > x_t$ (impact to right of structure)
	L	$x < x_t$ (impact to left of structure)
	A	$y > y_t$ (impact above structure)
	B	$y < y_t$ (impact below structure)

Orientation	θ_1	θ_2
+ R B	$\alpha - \beta$	$\alpha + \beta$
- R B	$-\alpha - \beta$	$-\alpha + \beta$
+ R A	$-\alpha - \beta$	$-\alpha + \beta$
+ L B	$-\alpha - \beta + \pi$	$-\alpha + \beta + \pi$
- L B	$\alpha - \beta - \pi$	$\alpha + \beta - \pi$
+ L A	$\alpha - \beta - \pi$	$\alpha + \beta - \pi$

If the crashed aircraft maintains a heading in the range $[\theta_1, \theta_2]$, given sufficient momentum it will slide into the structure with a velocity in excess of v_c . The relevant probability of having sufficient momentum is $P[D_s > d_e(x, y, \theta)]$, where D_s is the distance the aircraft would slide at horizontal velocity v_c or greater and the effective distance $d_e(x, y, \theta)$ is the distance the aircraft needs to slide to reach the structure. This probability is expressible in terms of variables which can be modeled from crash data. A computable expression for $d_e(x, y, \theta)$ must, however, be derived. The geometry of the slide is shown in Figure 6, where α denotes the heading of the aircraft relative to head on, i.e., $\alpha = \theta - (\theta_1 + \theta_2)/2$. A solution for d_e is obtained by solving simultaneously the three trigonometric equations,

$$\sin \alpha = z/d_e \quad (10)$$

$$\sin \beta = z/g \quad (11)$$

$$d_e^2 = d^2 + g^2 - 2dg \cos \beta, \quad (12)$$

where $g = r + w$. Substitution of $z = d_e \sin \alpha$ from (10) into (11) gives

$$\beta = \sin^{-1} \left(\frac{d_e}{g} \sin \alpha \right). \quad (13)$$

Substitution of (13) into (12) gives

$$d_e^2 = d^2 + g^2 - 2 d g \sqrt{1 - \left(\frac{d_e}{g} \sin \alpha \right)^2}, \quad (14)$$

since $\cos \sin^{-1} x = \sqrt{1-x^2}$. Define $\lambda = g/d$ and $\xi = (d_e/d)^2$. Then from (14),

$$2 \lambda \sqrt{1 - \frac{\xi}{\lambda^2} \sin^2 \alpha} = 1 + \lambda^2 - \xi, \quad (15)$$

and after squaring both sides and reducing,

Figure 6. The effective distance between the aircraft (large disk) and the structure (small disk) is d_e , if the aircraft is sliding at an angle α relative to head on.

$$\xi^2 - 2(\lambda^2 - 2\sin^2 \alpha + 1)\xi + (\lambda^2 - 1)^2 = 0. \quad (16)$$

Equation (16) may be solved for ξ :

$$\xi = \lambda^2 + 1 - 2\sin^2 \alpha \pm 2\cos \alpha \sqrt{\lambda^2 - \sin^2 \alpha}. \quad (17)$$

The negative root is appropriate, since for the head on case ($\alpha = 0$) this provides $d_e = d - g$. Therefore

$$d_e = d \sqrt{\lambda^2 + 1 - 2\sin^2 \alpha - 2\cos \alpha \sqrt{\lambda^2 - \sin^2 \alpha}}, \quad (18)$$

where $\lambda = (r + w)/d$ and $\alpha = \theta - (\theta_1 + \theta_2)/2$.

A usable expression for $P(D_s > d_e)$ can be developed from simple Newtonian principles. If the aircraft is assumed to slide after impact with constant deceleration $a > 0$, then the velocity v of the sliding aircraft as a

function of the distance s traveled along the ground after impact is given by the relationship

$$v^2 = v_0^2 - 2 a s, \quad (19)$$

where v_0 is the initial horizontal velocity,

$$v_0 = v_k \cos \phi, \quad (20)$$

v_k is the impact velocity, and ϕ is the impact angle. Stochastically, the deceleration A , which depends on the terrain, is independent of impact velocity and angle (V_k, Φ) , which depend on the aircraft's flight characteristics. Crash data suggest that V_k and Φ are dependent. The approach in expressing $P(D_s > d_e)$ is to obtain a range of deceleration values for which the event $D_s > d_e$ occurs and then obtain a range of impact velocity and angle values for each acceptable deceleration.

The event $D_s > d_e$ is equivalent to the velocity of the aircraft after sliding the distance d_e exceeding v_c , which from (19) is equivalent to the inequality

$$a < \frac{v_0^2 - v_c^2}{2d_e}. \quad (21)$$

It is assumed that $v_0 > v_c$; otherwise $P(D_s > d_e) = 0$. If the deceleration distribution is restricted to the range $[a, \bar{a}]$ and the impact velocity distribution to the range $[v_k, \bar{v}_k]$, then the deceleration range for which (21) may hold is

$$R_a \equiv \left[\min \left(\frac{\bar{v}_k^2 - v_c^2}{2d_e}, a \right), \min \left(\frac{\bar{v}_k^2 - v_c^2}{2d_e}, \bar{a} \right) \right]. \quad (22)$$

Consider a fixed value of deceleration, a . For $D_s > d_e$ to occur, it is necessary that

$$v_0^2 > v_c^2 + 2 a d_e. \quad (23)$$

The range of impact velocities for which (23) may hold is therefore

$$R_v(a) \equiv \left[\max\left(\sqrt{v_c^2 + 2ad_e}, \underline{v}_k\right), \max\left(\sqrt{v_c^2 + 2ad_e}, \bar{v}_k\right) \right]. \quad (24)$$

Finally, consider both deceleration a and impact velocity v_k fixed. Inequality

(23) is equivalent to $\phi < \cos^{-1}\left(\sqrt{\frac{v_c^2 + 2ad_e}{v_k^2}}\right)$. Consequently,

$$P(D_s > d_e) = \int_{R_a} f_A(a) da \int_{R_v(a)} f_{V_k}(v) F_{\Phi|V_k}\left(\cos^{-1}\left(\sqrt{\frac{v_c^2 + 2ad_e}{v^2}}\right) | v\right) dv, \quad (25)$$

where f_A and f_{V_k} are the pdfs of A and V_k , $F_{\Phi|V_k}$ is the conditional CDF of Φ given V_k , and the ranges R_a and $R_v(a)$ are defined in (22) and (24).

Although (25) is essentially a general result, its use is limited to commercial aircraft because certain aspects of military and general aviation data introduce incompatibility. In the military setting, a degeneracy in the distribution of deceleration has been noted. For sufficiently large impact angles there is effectively infinite deceleration, i. e., if Φ exceeds some threshold ϕ_c , the length of the skid is negligible. Thus for $D_s > d_e$ to occur, Φ must be less than the threshold ϕ_c , and conditional on this event, the horizontal velocity V_0 at impact must be sufficiently large (cf. (23)) for a given deceleration. Therefore,

$$P(D_s > d_e) = P(\Phi < \phi_c) \int_0^{\frac{v_0^2 - v_c^2}{2d_e}} f_{A|\Phi < \phi_c}(a) P\left(V_0 > \sqrt{v_c^2 + 2ad_e} | \Phi < \phi_c\right) da. \quad (26)$$

In the general aviation data base, the length of the skid has not been associated with the impact angle and impact velocity. As a consequence the deceleration distribution is not available. However, a regression model has been proposed by general aviation experts that provides the distribution of skid distance S as a function of impact angle ϕ , the impact velocity v_k , and the

weight of the aircraft, w . By applying (19) to $v=v_c$ and $v=0$, it is evident that S is related to D_s by

$$S = \frac{D_s}{1 - \left(\frac{v_c}{v_0}\right)^2}. \quad (27)$$

Therefore,

$$P(D_s > d_e) = \int_0^{\frac{\pi}{2}} \int_{\min(\bar{v}_k, \max(\underline{v}_k, \frac{v_c}{\cos\phi}))}^{\bar{v}_k} f_\phi(\phi) f_{v_k}(v) \left[1 - F_{S|\phi, v_k} \left(\frac{d_e}{1 - (v_c/v_0)^2} \right) \right] dv d\phi. \quad (28)$$

The assumption of statistical independence of impact angle and velocity has been made as a concession to the paucity of reliable data.

To conclude, $P(E \mid C_o)_s$ is obtained from (5), Table 1, (18), and (25), (26), or (28).

2.1.2. Obtaining an expression for $P(E \mid C_o)_g$.

A glancing impact with the structure occurs if the aircraft strikes the structure on its way down, although no impact would have occurred if the structure's height were 0. It is necessary to define the location of "conceptual impact" in this instance. This is where the aircraft would have impacted with the ground had there been no structure to interfere. In order for E to occur by a glancing hit the following sequence of events must take place:

- The conceptual impact must occur off the runway somewhere outside the circular direct hit region.
- The heading at conceptual impact must be in such a range that the aircraft can hit the structure on its way down.

- The conceptual impact angle must be sufficiently small that the aircraft is low enough to hit the structure on its way down.
- The conceptual impact velocity must exceed v_c .

The expression for $P(E \mid C_o)_g$ is similar to the slide case, (5). For a given orientation of structure and (conceptual) impact the essential geometric difference between a sliding and glancing hit is the direction of travel along the ground: one is opposite of the other. Consequently the heading ranges for glancing hits are those for sliding hits with an additive adjustment of $\pm\pi$. (Either sign is correct because of 2π periodicity.) For there to be impact with a structure of height h , the altitude of the aircraft must be no greater than h when it intersects the projected cylinder of the structure at the closest point to the conceptual impact. This is shown in Figure 6, where now the aircraft is traveling from the top of the figure to the bottom. The aircraft is assumed to be descending at the constant angle ϕ . Glancing impact occurs if the conceptual impact angle ϕ satisfies the inequality $\tan\phi \leq h/d_e$. See Figure 7.

Figure 7. The maximum glancing hit impact angle ϕ satisfies $\tan\phi = h/d_e$.

Hence $P(E \mid C_o)_g =$

$$\iint_{(C \cup R)} f_{X,Y}(x,y) \int_{\theta_1^*(x,y)}^{\theta_2^*(x,y)} f_{\Theta|X,Y}(\theta|x,y) \int_{\max(v_c, v_k)}^{v_k} f_{V_k}(v) F_{\Phi|V_k}(\tan^{-1}(h/d_e)|v) dv d\theta dy dx, \quad (29)$$

where the heading range $[\theta_1^*, \theta_2^*]$ is given in Table 2, d_e is given in (18) with $\alpha = \theta - (\theta_1^* + \theta_2^*)/2$, and the inner integral is the probability $P(\tan \Phi \leq h/d_e)$ obtained by conditioning on the impact velocity V_k .

Table 2. Heading ranges $[\theta_1^*, \theta_2^*]$ for glancing hits.

$$\begin{aligned} \text{Definitions: } \alpha &= \sin^{-1}\left(\frac{|y_t - y|}{d}\right) \\ \beta &= \sin^{-1}\left(\frac{w + r}{d}\right) \end{aligned}$$

Abbreviations:	CI	conceptual impact (x, y)
	+	y > 0 (CI above runway)
	-	y < 0 (CI below runway)
	R	x > x _t (CI to right of structure)
	L	x < x _t (CI to left of structure)
	A	y > y _t (CI above structure)
	B	y < y _t (CI below structure)

Orientation	θ_1^*	θ_2^*
+ R B	$\alpha - \beta - \pi$	$\alpha + \beta - \pi$
- R B	$-\alpha - \beta + \pi$	$-\alpha + \beta + \pi$
+ R A	$-\alpha - \beta + \pi$	$-\alpha + \beta + \pi$
+ L B	$-\alpha - \beta$	$-\alpha + \beta$
- L B	$\alpha - \beta$	$\alpha + \beta$
+ L A	$\alpha - \beta$	$\alpha + \beta$

2.1.3. Obtaining an expression for $P(E \mid C_o)_d$.

A direct impact with the structure occurs if the aircraft strikes the structure on its way down, and impact would have occurred even if the structure's height were 0. As in the glancing hit case the notion of conceptual impact is needed. In order for E to occur by a direct hit the following sequence of events must take place:

- The conceptual impact must occur off the runway somewhere within the circular direct hit region.
- The conceptual impact velocity must exceed v_c .

Therefore

$$P(E \mid C_o)_d = [1 - F_{V_k}(v_c)] \iint_{C \cap R'} f_{X,Y}(x,y) dy dx, \quad (30)$$

where F_{V_k} is the CDF of V_k .

2.2. Runway accidents.

Recall that a runway accident C_r is an overrun or run-off of some kind, where the aircraft exits the runway on the ground. Although typically the structure of interest will be located far enough from all runways that a collision resulting from a runway accident poses no credible threat, the situation is nonetheless modeled here for purposes of completeness.

Computation of $P(E \mid T, C_r)_{ji}$ and $P(E \mid L, C_r)_{ji}$, either of which for the sake of brevity will be denoted as $P(E \mid C_r)$, is similar in spirit to that of $P(E \mid C_o)_s$. In the slide scenario the ultimate stopping point is determined mathematically

by the impact point, heading, initial horizontal velocity, and deceleration. In an overrun/run-off the stopping point is determined by the point at which the aircraft exits the runway, along with the associated heading, velocity, and deceleration. A distinction is made whether the aircraft exits on the side or the end of the runway. Given a runway accident, the exit occurs on the upper side ($y=1$), the lower side ($y=-1$), or the left end ($x=-q$). Since the intended direction of travel is right to left, it is assumed inconceivable that the aircraft would leave the runway on the right end. Denote by p_{up} , p_{low} , and p_{end} the respective conditional probabilities of exiting on the upper side, lower side, and left end. Then

$$p_{up} + p_{low} + p_{end} = 1, \quad (31)$$

and by symmetry

$$p_{up} = p_{low}. \quad (32)$$

Relative frequency estimates of these probabilities are available from the accident data base. Since the structure is assumed to be located above the runway, the event E cannot occur in the context of a runway accident if a lower side run-off occurs. Thus

$$\begin{aligned} P(E \mid C_r) = & p_{up} \int_0^q g_{side}(x) dx \int_{\theta_1^{side}(x)}^{\theta_2^{side}(x)} g_{\theta|X}(\theta|x) P(D_s > d_e(x, 1, \theta)) d\theta \\ & + p_{end} \int_{-1}^1 g_{end}(y) dy \int_{\theta_1^{end}(y)}^{\theta_2^{end}(y)} g_{\theta|Y}(\theta|y) P(D_s > d_e(0, y, \theta)) d\theta, \end{aligned} \quad (33)$$

where $g_{side}(x)$ is the pdf of the x -coordinate of the exit point on the upper side of the runway; $g_{\theta|X}(\theta|x)$ is the conditional pdf of the aircraft's heading as it exits the runway at $(x, 1)$; $g_{end}(y)$ is the pdf of the y -coordinate of the exit point on the departure end of the runway; $g_{\theta|Y}(\theta|y)$ is the conditional pdf of the aircraft's heading as it exits the runway at $(0, y)$; $[\theta_1^{side}(x), \theta_2^{side}(x)]$ is the range of headings from the side exit point for which the aircraft given sufficient

momentum would ultimately slide into the structure; $[\theta_1^{\text{end}}(y), \theta_2^{\text{end}}(y)]$ is the range of headings from the end exit point for which the aircraft given sufficient momentum would ultimately slide into the structure; D_s is the length of the portion of the slide from the runway exit point during which the velocity exceeds v_c ; and $d_e(x, y, \theta)$ is the distance the aircraft must slide from the exit point (x, y) at heading θ in order to hit the structure.

The heading ranges can be extracted from Table 1. For a side exit, only the "+ R B", "+ L B", "+ R A", and "+ L A" entries are used, since the aircraft exits with $y = 1 > 0$. To ensure that the aircraft does not exit the end of the runway the additional restriction to $[0, \pi]$ is required. See Table 3. For an end exit, all the entries of Table 1 are used, with the added restriction to $[-\pi/2, \pi/2]$ so that the aircraft does not exit the side. See Table 4.

The probabilities $P[D_s > d_e(x, 1, \theta)]$ and $P[D_s > d_e(0, y, \theta)]$ of sliding far enough at velocities in excess of v_c are obtained as in Section 2.1.1. The effective distances d_e may be computed from (18). For commercial accidents the probability $P(D_s > d_e)$ is simpler than (25) because only the (horizontal) velocity V_r and deceleration A_r of the aircraft as it exits the runway are featured. Assume the distribution of V_r has range $[\underline{v}_r, \bar{v}_r]$ and the distribution of A_r has range $[\underline{a}_r, \bar{a}_r]$. Then

$$P(D_s > d_e) = \int_{R_a^*} g_{A_r}(a) da \int_{R_v^*(a)} g_{V_r}(v) dv, \quad (34)$$

where g_{A_r} and g_{V_r} are the pdfs of A_r and V_r ,

$$R_a^* = \left[\min\left(\frac{\bar{v}_r^2 - v_c^2}{2d_e}, \underline{a}_r\right), \min\left(\frac{\bar{v}_r^2 - v_c^2}{2d_e}, \bar{a}_r\right) \right], \quad (35)$$

and

$$R_v^*(a) = \left[\max\left(\sqrt{v_c^2 + 2ad_e}, \underline{v}_r\right), \max\left(\sqrt{v_c^2 + 2ad_e}, \bar{v}_r\right) \right]. \quad (36)$$

The result (34) is appropriate also for military and general aviation runway slides, although no data on such accidents have heretofore been analyzed to enable calculation of the pdfs of A_r and V_r .

Table 3. Heading ranges for slides into the structure from an upper side exit from the runway at (x, l) .

$$[\theta_1^{\text{side}}(x), \theta_2^{\text{side}}(x)] = [\theta_1, \theta_2] \cap [0, \pi].$$

Definitions: $\alpha = \sin^{-1}\left(\frac{|y_t - l|}{d}\right)$

$$\beta = \sin^{-1}\left(\frac{w + r}{d}\right)$$

Abbreviations:	+	$l > 0$ (exit above runway center line)
	R	$x > x_t$ (exit to right of structure)
	L	$x < x_t$ (exit to left of structure)
	A	$l > y_t$ (exit above structure)
	B	$l < y_t$ (exit below structure)

Orientation	θ_1	θ_2
+ R B	$\alpha - \beta$	$\alpha + \beta$
+ R A	$-\alpha - \beta$	$-\alpha + \beta$
+ L B	$-\alpha - \beta + \pi$	$-\alpha + \beta + \pi$
+ L A	$\alpha - \beta + \pi$	$\alpha + \beta + \pi$

Table-4. Heading ranges for slides into the structure from a left end exit from the runway at (0, y).

$$[\theta_1^{\text{end}}(y), \theta_2^{\text{end}}(y)] = [\theta_1, \theta_2] \cap [-\pi/2, \pi/2].$$

Definitions: $\alpha = \sin^{-1}\left(\frac{|y_t - y|}{d}\right)$

$$\beta = \sin^{-1}\left(\frac{w + r}{d}\right)$$

Abbreviations:

+	$y > 0$ (exit above runway center line)
-	$y < 0$ (exit below runway center line)
R	$-q > x_t$ (exit to right of structure)
L	$-q < x_t$ (exit to left of structure)
A	$y > y_t$ (exit above structure)
B	$y < y_t$ (exit below structure)

Orientation	θ_1	θ_2
+ R B	$\alpha - \beta$	$\alpha + \beta$
- R B	$-\alpha - \beta$	$-\alpha + \beta$
+ R A	$-\alpha - \beta$	$-\alpha + \beta$
+ L B	$-\alpha - \beta + \pi$	$-\alpha + \beta + \pi$
- L B	$\alpha - \beta - \pi$	$\alpha + \beta - \pi$
+ L A	$\alpha - \beta - \pi$	$\alpha + \beta - \pi$

2.3. Probability distributions and parameter settings used in applications.

Computation of the probabilities $P(E \mid C_o)_s$, $P(E \mid C_o)_g$, and $P(E \mid C_o)_d$ requires the complete specification of the probability distributions that model the impact location, angle, and velocity, and the heading and deceleration or skid distance after impact; computation of the probability $P(E \mid C_r)$ requires the complete specification of the probability distributions that model the overrun/run-off location, heading, velocity, and deceleration. Each distribution may in principle depend on the type of aircraft, the runway in question, and whether the accident involves a take-off or a landing. In most cases reasonable parametric families of distributions and associated parameter settings have been obtained based on the examination and analysis of large quantities of data extracted from published reports of commercial, military, and general aviation aircraft accidents. The commercial data base is essentially complete for large commercial jet accidents of the period from 1950 to 1990. The military data base covers mishaps between 1979 and 1993, and the general aviation data base concerns the period from 1986 to 1992. In some military and general aviation cases a nonparametric approach has been taken in response to scant or unreliable data. The runway is a contributor to the models only through its role in determining the coordinate system. Locations are measured in reference to the runway; only the runway's length and width are pertinent. The prototypical runway with length $q=10000$ feet and width 150 feet ($l=75$) is used in all distributions presented below.

Most quantities modeled have finite ranges. Beta distributions have been successfully applied in cases of limited ranges, e.g. heading angles and positions on the runway's edge. In cases of finite but essentially open-ended variables, e.g. impact locations, velocities, and decelerations, truncated

versions of common distributions with infinite range (lognormal, Weibull, and Pearson) have been successfully fitted to the data.

All results should be viewed as preliminary. Additions and revisions of data will undoubtedly lead to improved parameter estimates.

The units used are (feet, seconds, radians) for (distance, time, angle).

2.3.1. Off-runway impact location distributions.

Consider the location of an impact caused by a crash on take-off. Recall that the x -coordinate measures the position along the extended runway center line, the intended direction of travel is right to left, and the origin is at the right extreme (threshold end) of the runway. In the commercial setting the joint pdf of (X, Y) used in equations (5), (29), and (30) is conveniently expressed as

$$f_{X,Y}(x, y) = f_X(x) f_{Y|X}(y | x), \quad (37)$$

where $f_X(x)$ denotes the pdf of the x -coordinate, and $f_{Y|X}(y | x)$ denotes the conditional pdf of the y -coordinate given the x -coordinate. The commercial data base shows no accidents with x -coordinate larger than 0, as this would require the aircraft to impact at a point behind where it commenced its take-off run down the runway. In addition, all impacts have x -coordinates greater than -90000 (about 15 miles beyond the departure end of the runway), since impacts less than this figure can occur only if the aircraft is beyond the take-off/climb phase of flight. A truncated lognormal distribution has been found to adequately fit the x -coordinate of take-off impacts. Specifically, the pdf $f_X(x)$ is given by

$$f_X(x) = \frac{h(x)}{1 - H(x_1)}, \quad (38)$$

$$\text{where } h(x) = \frac{1}{-x\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}[\ln(-x) - \mu]^2\right), -\infty < x < 0, \quad (39)$$

$$\text{and } H(t) = \int_{-\infty}^t h(x) dx, \quad (40)$$

with $x_1 = -90000$, $\mu = 9.3663$, and $\sigma = 0.4913$. Essentially $\ln(-X)$ is normally distributed with mean 9.3663 and standard deviation 0.4913, with truncation on the left at $x = -90000$. Hence the mean of the distribution of X is around -13200, implying that a typical impact occurs about 3200 feet beyond the departure end of the runway. See Figure 8. The distribution of the y-coordinate of impact depends on the value of the x-coordinate, since the distance from the extended runway center line tends to increase as the flight progresses and a jetway is followed toward a destination. A Weibull log linear regression that employs maximum likelihood techniques on exact, censored, and interval data has produced a satisfactory fit. The conditional pdf $f_{Y|X}(y|x)$ is determined as follows:

- $Y | x$ is distributed as $W_x + 75$ with probability 1/2, and as $-W_x - 75$ with probability 1/2, if $-10000 \leq x \leq 0$
- $Y | x$ is distributed as W_x with probability 1/2, and as $-W_x$ with probability 1/2, if $-90000 \leq x < -10000$
- W_x is Weibull with scale parameter $\alpha(x) = \exp(4.9576 - 0.00009955 x)$ and shape parameter $c = 0.5381$, i.e., W_x has pdf

$$g(w) = \frac{c}{\alpha(x)} \left(\frac{w}{\alpha(x)}\right)^{c-1} \exp\left(-\left[\frac{w}{\alpha(x)}\right]^c\right). \quad (41)$$

A feature of the dependency of Y on x is shown in Figure 9 in which the median y-coordinate is plotted against x for impacts above the runway. The

mirror image is valid for impacts below the runway, as the impacts in the commercial data base occur with probabilistic symmetry about the extended runway center line. Actually each individual airport has established flight patterns which induce an asymmetry in the frequency of activity in the $y > 0$ versus the $y < 0$ regions. However, commercial crash reports do not reference accidents with respect to patterns, and the totality of impacts over all airports average out to create symmetry about the x-axis.

Figure 8. The truncated lognormal pdf for the x-coordinate of commercial off-runway take-off crashes.

Nonparametric kernel estimation procedures have been implemented to provide joint pdfs for the military impact locations. Parametric models do not appear to be justified because of the scantness of the data. The kernel densities are not symmetric with respect to the x-axis. There is a strong tendency for military aircraft to impact on the pattern side of the runway, and,

Figure 9. The median y-coordinate of impact for commercial off-runway take-off crashes above the runway.

unlike commercial accident reports, military accident reports specify the location in terms of the pattern. Without loss of generality the pattern side is defined as $y > 0$. Since the facility of interest has by convention a positive y-coordinate, it is necessary computationally to flip the kernel density about the x-axis whenever the facility is actually located on the non-pattern side. The kernel densities may be approximated by using values at selected points in a grid. The range of coordinates having positive density values for take-off impacts is $-45440 \leq x \leq 12640$, $-13200 \leq y \leq 55440$.

Similar nonparametric kernel estimation techniques have also been invoked for general aviation impact locations. Parametric modeling appears difficult because of the poor precision of the data. The kernel densities may be approximated by using values at selected points in a grid. The range of

coordinates having positive density values for impacts is $-47800 \leq x \leq 57800$, $-52800 \leq y \leq 52800$.

Modeling the location of off-runway landing impacts is analogous to the take-off situation. Landing phase crashes tend to occur short of the runway. Since no commercial data base crashes occurred beyond the runway departure end, and crashes with x-coordinate greater than 105000 (about 20 miles before the runway threshold) are not viewed as landing/landing approach phase accidents, an x-coordinate range from -10000 to 105000 seems appropriate in the commercial setting. A truncated lognormal distribution provides an adequate fit for the x-coordinate of landing impacts. Specifically, the pdf $f_X(x)$ used in equation (37) is given by

$$f_X(x) = \frac{k(x)}{K(x_u)}, \quad (42)$$

$$\text{where } k(x) = \frac{1}{(x - x_0)\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\{\ln[x - x_0] - \mu\}^2\right), \quad x > x_0, \quad (43)$$

$$\text{and } K(t) = \int_{x_0}^t k(x) dx, \quad (44)$$

with $x_u = 105000$, $x_0 = -10000$, $\mu = 9.6607$, and $\sigma = 0.4947$. Thus $\ln(X + 10000)$ is normally distributed with mean 9.6607 and standard deviation 0.4947, with truncation on the right at $x = 105000$. Hence the mean of the distribution of X is around 7600, implying that a typical impact occurs about 7600 feet short of the threshold end of the runway. See Figure 10. The distribution of the y-coordinate of impact depends on the x-coordinate, since typical landing patterns involve turning toward the extended runway center line. The Weibull log linear regression methodology employed for the take-off data has been used to fit the landing data with comparable success. Landing phase impacts tend to fall closer to the extended runway center line as the flight

Figure 10. The truncated lognormal pdf for the x-coordinate of commercial off-runway landing crashes.

progresses and the approach to the runway is executed. The conditional pdf $f_{Y|X}(y|x)$ used in equations (5), (29), and (30) is determined as follows:

- $Y | x$ is distributed as W_x with probability $1/2$, and as $-W_x$ with probability $1/2$, if $0 \leq x \leq 105000$
- $Y | x$ is distributed as $W_x + 75$ with probability $1/2$, and as $-W_x - 75$ with probability $1/2$, if $-10000 \leq x \leq 0$
- W_x is Weibull with scale parameter $\alpha(x) = \exp(5.8542 + 0.000056063 x)$ and shape parameter $c = 0.82406$, i.e., W_x has pdf (41).

In Figure 11 the median y-coordinate is plotted against x for impacts above the runway. The mirror image is valid for impacts below the runway.

As in the take-off case, nonparametric kernel estimators have been obtained for military and general aviation landing impact locations. Approximations are again available based on values of the density at selected grid points. The range of coordinates having positive density values for

military landing impacts is $-2640 \leq x \leq 81840$, $-29040 \leq y \leq 55440$. The range of coordinates having positive density values for general aviation impacts is $-47800 \leq x \leq 57800$, $-52800 \leq y \leq 52800$.

Figure 11. The median y-coordinate of impact for commercial off-runway landing crashes above the runway.

2.3.2. Off-runway heading distributions.

The aircraft's heading after an off-runway impact is not explicitly given in accident reports. The resting location of the wreckage is however reported, at least for commercial and military accidents, and the heading is estimated by calculating the angle, relative to the extended runway center line, of the line from the impact point to the center of the main wreckage segment.

The heading of the aircraft after a take-off crash depends on the impact location. Because typical take-off flight patterns tend to require a turn away

from the extended runway center line, the heading is usually greater than the arc tangent of the position of the aircraft. See Figure 12. A regression on

Figure 12. The heading θ of the aircraft on take-off is typically greater than the arc tangent $\theta_0 = \tan^{-1}(|y|/|x|)$ of its position.

headings in commercial off-runway take-off impacts has found the mean heading θ_0^* to satisfy the log linear relation

$$\theta_0^* = 0.97126 + 0.16906 \ln(\theta_0), \quad (45)$$

where θ_0 marks the nominal heading on impact, i.e.,

$$\theta_0 = \tan^{-1}(|y|/|x|), \quad (46)$$

and the heading itself to be fit adequately as

$$\Theta = \theta_0^* + 2\pi \left(B - \frac{1}{2} \right), \quad (47)$$

where B denotes a symmetric beta distributed random variable with parameter 132. That is, B has pdf

$$g_B(x; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}, 0 < x < 1, \quad (48)$$

with $a = b = 132$. Therefore the conditional pdf $f_{\Theta|X,Y}$ used in equations (5) and (29) for commercial accidents is given by

$$f_{\Theta|X,Y}(\theta | x, y) = \frac{1}{2\pi} g_B\left(\frac{\theta - \theta_0^* + \pi}{2\pi}; 132, 132\right), \theta_0^* - \pi < \theta < \theta_0^* + \pi. \quad (49)$$

The mean heading θ_0^* as a function of nominal heading θ_0 is shown in Figure 13.

Figure 13. The mean heading, in degrees, at impact for commercial off-runway take-off crashes as a function of the nominal heading, in degrees.

The headings in off-runway landing impacts do not appear from the commercial data base to depend on the location. A beta distribution with a modest tendency for headings away from the extended runway center line provides a satisfactory fit. Specifically, Θ is distributed as

$$\Theta = 2\pi \left(B - \frac{1}{2} \right), \quad (50)$$

where B has pdf (48) with $a=61.18$ and $b=56.76$, so that the conditional pdf $f_{\Theta|X,Y}$ used in equations (5) and (29) for commercial accidents is given by

$$f_{\Theta|X,Y}(\theta | x, y) = \frac{1}{2\pi} g_B\left(\frac{\theta + \pi}{2\pi}; 61.18, 56.76\right), -\pi < \theta < +\pi. \quad (51)$$

This density is plotted in Figure 14. The mean value is 6.75 degrees.

Figure 14. The beta pdf of heading, in degrees, for commercial off-runway landing crashes.

Heading angles for military accidents have been modeled by normal distributions truncated to $(-\pi, \pi)$. Actually, military headings are defined in relation to the intended line of flight rather than the runway center line. However, since it is impractical if not impossible calculationally to locate the structure of interest relative to each conceivable intended flight pattern, the military line of flight heading angle distributions are used in the computer code as if they were defined in relation to the runway center line. The take-off headings for large military aircraft are modeled then as truncated normal with mean $\mu = -0.2524$ and standard deviation $\sigma = 1.046$; for small military aircraft, the parameters are $\mu = -0.06288$ and $\sigma = 1.842$. For landing headings, the parameters for large military aircraft are $\mu = 0.2449$ and $\sigma = 0.5327$; for small military aircraft they are $\mu = 0.09721$ and $\sigma = 1.107$. The truncated normal pdf is given by

$$f_{\Theta}(\theta) = \frac{\frac{1}{\sigma} \phi\left(\frac{\theta - \mu}{\sigma}\right)}{\Phi\left(\frac{\pi - \mu}{\sigma}\right) - \Phi\left(\frac{-\pi - \mu}{\sigma}\right)}, \quad (52)$$

where

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right) \quad (53)$$

and

$$\Phi(z) = \int_{-\infty}^z \phi(t) dt. \quad (54)$$

The military heading angle pdfs for take-off crashes are shown in Figures 15, 16, 17, and 18.

Figure 15. The truncated normal pdf of heading, in degrees, for large military aircraft off-runway take-off crashes.

Figure 16. The truncated normal pdf of heading, in degrees, for small military aircraft off-runway take-off crashes.

Figure 17. The truncated normal pdf of heading, in degrees, for large military aircraft off-runway landing crashes.

Figure 18. The truncated normal pdf of heading, in degrees, for small military aircraft off-runway landing crashes.

General aviation accident reports typically do not give both impact and resting locations. Hence heading angles are not estimable. Commercial landing heading distributions are used as surrogates in the computer code.

2.3.3. Off-runway impact velocity, impact angle, and deceleration distributions.

Although impact velocities and angles are often provided in accident reports, decelerations after impact are not. Nonetheless, if the resting point of the main wreckage is reported, the deceleration rate may be calculated from Newtonian principles. According to equations (19) and (20) with $v = 0$, the deceleration, assuming a constant rate along the straight line segment of length s from the impact point to the resting point, is given by

$$a = \frac{v_k^2 \cos^2 \phi}{2s}. \quad (55)$$

In commercial accidents impact velocities and angles as well as decelerations do not appear to depend on whether the accidents are take-off or landing related. The impact velocities are adequately modeled by a truncated Weibull distribution. Specifically, the off-runway impact velocity pdf f_{v_k} of equations (25) and (29) is given for commercial accidents by

$$f_{v_k}(v) = \frac{c}{\alpha} \left[\frac{v - \underline{v}_k}{\alpha} \right]^{c-1} \exp \left(- \left[\frac{v - \underline{v}_k}{\alpha} \right]^c \right) / \left[1 - \exp \left(- \left[\frac{\bar{v}_k - \underline{v}_k}{\alpha} \right]^c \right) \right], \underline{v}_k < v < \bar{v}_k, \quad (56)$$

where $\alpha = 161.7$, $c = 1.951$, and $(\underline{v}_k, \bar{v}_k) = (100., 500.)$. The mean of this distribution is about 243. See Figure 19. Impact angles depend on impact velocities as larger angles are associated with both smaller and larger velocities. Weibull log linear regression provides a satisfactory fit to allow evaluation of the commercial conditional CDF $F_{\Phi | v_k}$ in equations (25) and (29). In fact, $\Phi | v$ is distributed as a truncated Weibull with pdf

$$g(\phi) = \frac{c}{\alpha(v)} \left(\frac{\phi}{\alpha(v)} \right)^{c-1} \exp \left(- \left[\frac{\phi}{\alpha(v)} \right]^c \right) / \left[\exp \left(- \left[\frac{\phi}{\alpha(v)} \right]^c \right) - \exp \left(- \left[\frac{\bar{\phi}}{\alpha(v)} \right]^c \right) \right], \phi < \phi < \bar{\phi}, \quad (57)$$

Figure 19. The truncated Weibull pdf of impact velocity for commercial off-runway take-off or landing crashes.

where $c = 1.08931$, $\alpha(v) = \exp(1.8580 - 0.022191 v + 0.000036529 v^2)$, and $(\underline{\phi}, \bar{\phi}) = \left(\frac{\pi}{360}, \frac{\pi}{2} \right)$, i.e., the range is from $1/2$ to 90 degrees. The median impact angle as a function of impact velocity is shown in Figure 20.

Figure 20. The median impact angle in degrees as a function of impact velocity for commercial off-runway take-off or landing crashes.

The calculated commercial decelerations are adequately modeled by a truncated lognormal distribution. Specifically, the off-runway deceleration pdf f_A of equation (25) is given by

$$f_A(a) = \frac{\frac{1}{a\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(\ln a - \mu)^2\right]}{\Phi\left(\frac{\ln \bar{a} - \mu}{\sigma}\right) - \Phi\left(\frac{\ln \underline{a} - \mu}{\sigma}\right)}, \quad \underline{a} \leq a \leq \bar{a},$$

$$f_A(a) = \frac{\frac{1}{a\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(\ln a - \mu)^2\right]}{\Phi\left(\frac{\ln \bar{a} - \mu}{\sigma}\right) - \Phi\left(\frac{\ln \underline{a} - \mu}{\sigma}\right)}, \quad \underline{a} \leq a \leq \bar{a}, \quad (58)$$

where $\mu = 3.0390$, $\sigma = 0.6770$, $(\underline{a}, \bar{a}) = (1., 150.)$, and Φ is the standard normal CDF,

$$\Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) dt. \quad (59)$$

The mean deceleration is approximately 26. See Figure 21.

Figure 21. The truncated lognormal pdf of deceleration for commercial off-runway take-off or landing crashes.

Equation (25), the commercial formulation of $P(D_s > d_e)$ based on a finite range of decelerations, is inapplicable for military accidents because the military data base shows no appreciable skid for sufficiently large impact angles, specifically $\phi \geq \phi_c = 0.349$ (i.e., 20°). Calculation of the military version, equation (26), requires the distribution of impact angle Φ , the conditional pdf of deceleration A given $\Phi < \phi_c$, and the conditional distribution of initial horizontal velocity V_0 given $\Phi < \phi_c$. The latter two distributions are derivable from accident data restricted by $\phi < \phi_c$, and the use of (55) and (20).

The impact angle Φ for military take-off accidents is modeled adequately by a beta distribution over $(0, \pi/2)$ whose parameters depend on the type of aircraft, large or small. For landing accidents analogous lognormal distributions truncated to $(0, \pi/2)$ are used. Thus for take-offs,

$$f_\Phi(\phi) = \frac{2}{\pi} g_B\left(\frac{2}{\pi} \phi; a, b\right), 0 < \phi < \frac{\pi}{2}, \quad (60)$$

(cf. (48)) where $(a, b) = (0.42, 2.76)$ for large military aircraft, and $(a, b) = (0.55, 1.24)$ for small military aircraft. See Figures 22 and 23. The respective mean values are approximately 12° and 28° . Likewise for landings,

$$f_\Phi(\phi) = k(\phi)/K\left(\frac{\pi}{2}\right), 0 < \phi < \frac{\pi}{2}, \quad (61)$$

(cf. (42), (43), and (44)), where $x_0 = 0$ in (43) and $(\mu, \sigma) = (-0.958, 1.61)$ for large military aircraft, and $(\mu, \sigma) = (-1.27, 1.66)$ for small military aircraft. See Figures 24 and 25. The respective mean values are approximately 23° and 20° . The probabilities $P(\Phi < \phi_c)$ in (26) are computed from integration involving (60) and (61) to be $(0.7824, 0.4928, 0.5889, 0.6491)$ for take-off large, take-off small, landing large, and landing small, respectively. The distributions (60) and (61) are also used in the calculation of equation (29).

Figure 22. The beta pdf of impact angle, in degrees, for large military aircraft off-runway take-off crashes.

Figure 23. The beta pdf of impact angle, in degrees, for small military aircraft off-runway take-off crashes.

Figure 24. The truncated lognormal pdf of impact angle, in degrees, for large military aircraft off-runway landing crashes.

Figure 25. The truncated lognormal pdf of impact angle, in degrees, for small military aircraft off-runway landing crashes.

The conditional distribution of deceleration given an impact angle below 20° is modeled adequately by a lognormal model. That is,

$$f_{A | \Phi < \phi_c}(a) = \frac{1}{a\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(\ln a - \mu)^2\right], a > 0, \quad (62)$$

where the parameters μ and σ are (4.4, 1.0), (4.9, 1.4), (3.8, 0.9), and (4.2, 1.1) for military take-off large, take-off small, landing large, and landing small, respectively. The respective means are approximately 134, 358, 67, and 122. The pdfs are plotted in Figures 26 through 29.

The conditional distribution of initial horizontal velocity, given an impact angle below 20° , is modeled adequately by a Pearson V model. Here

$$f_{V_0 | \Phi < \phi_c}(v) = \frac{\left(\frac{v}{\beta}\right)^{-(\alpha+1)} e^{-\beta/v}}{\beta \Gamma(\alpha)}, v > 0, \quad (63)$$

where the parameters α and β depend on the phase and the type of aircraft. For large military aircraft on take-offs, $(\alpha, \beta) = (50.47, 10800.)$; for small take-offs, $(\alpha, \beta) = (6.32, 1207.)$; for large landings, $(\alpha, \beta) = (9.48, 2245.)$; and for

small landings, $(\alpha, \beta) = (17.37, 4439.)$. The respective mean velocities are approximately 218, 227, 265, and 271. The pdfs are plotted in Figures 30 through 33.

Figure 26. The lognormal pdf of deceleration, given an impact angle below 20° , for large military aircraft off-runway take-off crashes.

Figure 27. The lognormal pdf of deceleration, given an impact angle below 20° , for small military aircraft off-runway take-off crashes.

Figure 28. The lognormal pdf of deceleration, given an impact angle below 20° , for large military aircraft off-runway landing crashes.

Figure 29. The lognormal pdf of deceleration, given an impact angle below 20° , for small military aircraft off-runway landing crashes.

Figure 30. The Pearson V pdf of initial horizontal velocity, given an impact angle below 20° , for large military aircraft off-runway take-off crashes.

Figure 31. The Pearson V pdf of initial horizontal velocity, given an impact angle below 20° , for small military aircraft off-runway take-off crashes.

Figure 32. The Pearson V pdf of initial horizontal velocity, given an impact angle below 20° , for large military aircraft off-runway landing crashes.

Figure 33. The Pearson V pdf of initial horizontal velocity, given an impact angle below 20° , for small military aircraft off-runway landing crashes.

The distribution of impact velocity is required for calculation of equations (29) and (30). For military take-offs a Pearson VI distribution is an adequate model. Thus

$$f_{V_k}(v) = \frac{\Gamma(\alpha_1 + \alpha_2) \left(\frac{v}{\beta}\right)^{\alpha_1 - 1}}{\beta \Gamma(\alpha_1) \Gamma(\alpha_2) (1 + v/\beta)^{\alpha_1 + \alpha_2}}, \quad v > 0, \quad (64)$$

where the parameters α_1 , α_2 , and β depend on the type of aircraft. For large military aircraft, $(\alpha_1, \alpha_2, \beta) = (160.0, 16.29, 24.49)$; for small military aircraft, $(\alpha_1, \alpha_2, \beta) = (36.15, 7.11, 66.70)$. The respective mean velocities are approximately 256 and 395. The pdfs are plotted in Figures 34 and 35. A Pearson V distribution (cf. (63)) is a satisfactory model for the impact velocity in military landing crashes. The parameters α and β are $(\alpha, \beta) = (5.81, 1440.)$ for large military aircraft, and $(10.0, 2903.)$ for small. The respective mean velocities are approximately 299 and 323. See Figures 36 and 37.

Figure 34. The Pearson VI pdf of impact velocity for large military aircraft off-runway take-off crashes.

Figure 35. The Pearson VI pdf of impact velocity for small military aircraft off-runway take-off crashes.

Figure 36. The Pearson V pdf of impact velocity for large military aircraft off-runway landing crashes.

Figure 37. The Pearson V pdf of impact velocity for small military aircraft off-runway landing crashes.

Calculation of equations (28), (29), and (30) for general aviation accidents requires distributions for impact velocity and impact angle. Analysis of the rather poor quality general aviation data base provides crude representations of the pdfs of V_k and Φ as independent histograms. For all types of general aviation aircraft and each phase, take-off and landing, the impact velocity is modeled by the following intervals and corresponding probabilities:

Velocity Interval (knots)	Probability
0 - 30	0.063
30 - 60	0.297
60 - 90	0.384
90 - 120	0.142
120 - 150	0.114.

Conversion to ft/sec is accomplished by multiplying the number of knots by 1.687806. For all types of general aviation aircraft and each phase, the impact angle is modeled by the following intervals and corresponding probabilities:

Impact Angle (degrees)	Probability
0 - 10	0.426
10 - 20	0.190
20 - 30	0.112
30 - 60	0.149
60 - 90	0.123.

Conversion to radians is accomplished by multiplying the number of degrees by $\pi/180 = 0.01745329$.

The skid distance distribution used in equation (28) is obtained by multiple regression as lognormal (cf. (62)) with parameters which may depend on the impact velocity, impact angle, and weight of the aircraft. General aviation experts have proposed the model,

$$\mu = \mu_0 + c_v + c_\phi + c_w \ln w, \quad (65)$$

and

$$\sigma = \sigma_0, \quad (66)$$

where c_v is a function of the impact velocity, c_ϕ is a function of the impact angle, w is the aircraft weight, and μ_0 , σ_0 , and c_w are parameters. The experts' current recommendation, however, is to collapse the effects of impact velocity, impact angle, and aircraft weight into an average effect represented by $\mu_0 = 2.9546$, $c_v \equiv 0$, $c_\phi \equiv 0$, $c_w = 0$, and $\sigma_0 = 1.513$. This gives $\mu \equiv 2.9546$ and $\sigma \equiv 1.513$, which generates a mean skid distance of around 60 feet.

2.3.4. - Runway run-off and overrun distributions and parameters.

Runway accidents in which the aircraft runs off the side of the runway or overruns the end of the runway and strikes the structure with sufficient velocity are characterized by equations (33) and (34). Only accidents from the commercial data base have heretofore been analyzed. The computer code uses the commercial parameters as surrogates for military and general aviation runway accidents. The parameter p_{end} is estimated from the relative frequency of end-related runway crashes in the commercial data base to be

$$p_{\text{end}} = 0.57. \quad (67)$$

Consequently, from (31) and (32),

$$p_{\text{up}} = 0.215. \quad (68)$$

The exit point (X, l) for a take-off related side run-off is adequately modeled from the data base as

$$X = 10000 (B - 1), \quad (69)$$

where B has the beta pdf (48) with parameters $a=1.400$ and $b=1.887$. The mean exit point is about 4300 feet from the departure end of the runway. See Figure 38. The pdf $g_{\text{side}}(x)$ of equation (33) for take-offs, then, is given by

$$g_{\text{side}}(x) = \frac{1}{10000} g_B\left(\frac{x}{10000} + 1; 1.400, 1.887\right), -10000 < x < 0, \quad (70)$$

where g_B is defined in (48). The exit point for a landing related side run-off is similarly modeled as (69), but with parameters $a=2.301$ and $b=1.552$. The mean exit point here is about 4000 feet from the threshold end of the runway. See

Figure 38. The beta pdf of take-off related side run-off exit point.

Figure 39. The beta pdf of landing related side run-off exit point.

Figure 39. The pdf $g_{\text{side}}(x)$ of equation (33) for landings is given by

$$g_{\text{side}}(x) = \frac{1}{10000} g_B\left(\frac{x}{10000} + 1; 2.301, 1.552\right), -10000 < x < 0. \quad (71)$$

The exit point $(-q, Y)$ for overruns from the departure end of the runway does not appear from the data base to depend on whether the accident is take-off or landing related. Furthermore, the distribution is satisfactorily modeled as a symmetric beta about the origin,

$$Y = 150 \left(B - \frac{1}{2}\right), -75 < y < 75, \quad (72)$$

where 150 is the runway width and B has the beta pdf (48) with parameters $a=b=2.317$. The pdf $g_{\text{end}}(y)$ of equation (33) for take-offs or landings is given by

$$g_{\text{end}}(y) = \frac{1}{150} g_B\left(\frac{y+75}{150}; 2.317, 2.317\right), -75 < y < 75. \quad (73)$$

See Figure 40.

The calculated heading of the aircraft as it exits the runway does not appear to depend on whether the accident is take-off or landing related. However, the angle does depend on whether the exit is from the side or the end, although the actual position along the side (x, l) or end $(-q, y)$ is not relevant. The heading in the case of side run-offs is adequately modeled by a beta distribution,

$$\Theta = \pi B, \quad (74)$$

where B is beta distributed (44) with parameters $a = 1.158$ and $b = 19.26$. The

Figure 40. The symmetric beta pdf of take-off or landing related end overrun exit point.

mean of this distribution is about 10 degrees. The pdf $g_{\Theta|X}$ of equation (33) is given by

$$g_{\Theta|X}(\theta | x) = \frac{1}{\pi} g_B\left(\frac{\theta}{\pi}; 1.158, 19.26\right), 0 < \theta < \pi. \quad (75)$$

See Figure 41. The fitted model for the heading in the case of end overruns is

$$\Theta = \pi (B - 1/2), \quad (76)$$

where now B is beta with parameters $a = 104.0$ and $b = 96.91$. The mean heading is about 3 degrees. The pdf $g_{\Theta|Y}$ of equation (33) is given by

$$g_{\Theta|Y}(\theta | y) = \frac{1}{\pi} g_B\left(\frac{\theta + \pi/2}{\pi}; 104.0, 96.91\right), -\frac{\pi}{2} < \theta < \frac{\pi}{2}. \quad (77)$$

See Figure 42.

Figure 41. The beta pdf of heading, in degrees, for take-off or landing side run-offs.

Figure 42. The beta pdf of heading, in degrees, for take-off or landing end overruns.

Exit velocities and decelerations do not appear to depend on whether the accidents are take-off or landing related. The exit velocities are adequately modeled by a truncated Weibull distribution. Specifically, the exit velocity pdf g_{v_r} of equation (34) is given by

$$f_{v_r}(v) = \frac{c}{\alpha} \left[\frac{v - \underline{v}_r}{\alpha} \right]^{c-1} \exp \left(- \left[\frac{v - \underline{v}_r}{\alpha} \right]^c \right) / \left[1 - \exp \left(- \left[\frac{\bar{v}_r - \underline{v}_r}{\alpha} \right]^c \right) \right], \quad \underline{v}_r < v < \bar{v}_r, \quad (78)$$

where $\alpha = 137.8$, $c = 1.53$, and $(\underline{v}_r, \bar{v}_r) = (10., 350.)$. The mean of this distribution is about 134. See Figure 43. Similarly, the exit deceleration has pdf g_{A_r}

Figure 43. The truncated Weibull pdf of exit velocity for runoff/overrun take-offs or landings.

satisfactorily modeled by the truncated lognormal,

$$f_{A_r}(a) = \frac{\frac{1}{a\sqrt{2\pi}\sigma} \exp \left[-\frac{1}{2\sigma^2} (\ln a - \mu)^2 \right]}{\Phi \left(\left[\ln \bar{a}_r - \mu \right] / \sigma \right) - \Phi \left(\left[\ln \underline{a}_r - \mu \right] / \sigma \right)}, \quad \underline{a}_r \leq a \leq \bar{a}_r, \quad (79)$$

where $\mu = 2.2897$, $\sigma = 0.6134$, $(\underline{a}_r, \bar{a}_r) = (1., 40.)$, and Φ is the standard normal CDF (59). The mean deceleration is approximately 12. See Figure 44.

Figure 44. The truncated lognormal pdf of exit deceleration for runoff/overrun take-offs or landings.

3. Limitations and Caveats.

The aircraft accident model for crashes in the vicinity of an airport uses a number of simplifying assumptions that are made primarily for mathematical convenience and yet are also in a sense the inevitable byproducts of a limited data base that for most accidents furnishes only a scant treatment of the flight and crash kinematics. So it is assumed that

- the slide after impact or runway exit follows a straight line path at constant deceleration
- the aircraft remains intact after impact
- the descent of the aircraft near impact is at a constant rate equal to the impact angle.

The first assumption is needed to calculate from Newtonian principles the velocity as a function of distance traveled after impact. In practice such things as intervention of the pilot or terrain irregularities could cause this assumption to be violated. The second assumption is needed to determine the range of headings of the slide that will allow collision with the structure. In many crashes there is fragmentation and scattering of segments of the aircraft. The resultant heading range, though wider than the intact model's, includes spacings wherein no fragment can slide into the structure. Modeling of the spacings from the accident reports appears difficult. The intactness assumption is a compromise that provides a contiguous range of appropriate size. The third assumption is needed to determine if a glancing hit occurs. Pilot avoidance maneuvers would cause this assumption to be violated as well as affect the direct hit modeling.

The assumptions that the aircraft is a disk with diameter equal to its wingspan and that the structure is a cylinder are also motivated by considerations of mathematical convenience. The objective is to estimate the probability that an aircraft would strike the structure with sufficient velocity. For this purpose it does not seem critical to model the shape of the aircraft and structure precisely; rather it appears satisfactory to use a disk and cylinder with representative sizing.

In applying the model to compute $P_W(E)$ for a particular structure, the quality of the estimate is influenced by the quality and appropriateness of the fitted probability distributions of Section 2.3. Relevant issues are

- the fitted distributions are the result of a statistical pooling of data over a large number of accidents involving a large number of aircraft types, airport configurations, flight patterns, terrains, weather conditions, and so

on; the structure and associated airport of interest have a much more limited set of potential accident characteristics

- the parametric and nonparametric models used are not unique; alternative models may provide improved descriptions of the underlying processes
- the quality of the data extracted from accident reports is questionable.

The first issue addresses the problem that the fitted model may not be pertinent in a given application. Pooling of data from many sources is made necessary by the rareness of accidents. This results, however, in fitted distributions that as products of an averaging process do not necessarily adequately describe a particular airport/structure setup. Thus for example the fitted commercial distribution of Y given x is symmetric about $y=0$, the extended runway center line; the take-off and landing patterns at a given airport may not suggest this symmetry. Likewise the fitted deceleration distribution may be inappropriate for a given airport's local terrain. In applications where the fitted distribution is clearly unrepresentative it is reasonable to substitute an alternative distribution customized to characterize the local setup. The second issue concerns the influence of the data base on model selection. The parametric models that are presented in Section 2.3 represent good statistical fits of the existing data. Other models may achieve comparable statistical fits and still others may for theoretical reasons better describe the true underlying accident mechanisms. The third issue acknowledges the relatively poor quality of the data base. Much of the data had to be extracted from reports which contained ambiguities, contradictions, omissions, and a general lack of precision. As a consequence the analysis frequently uses intervals rather than point values for measured quantities.

4. Computer Codes.

Fortran codes have been written by the author for execution on Macintosh and IBM compatible personal computers to

- compute the probability $P_W(E)$
- implement Weibull log linear regression.

Distribution fitting for non-Weibull models can be carried out with standard statistical software packages such as Systat, SAS, or Stata.

The code for computation of $P_W(E)$ is given below. It is in a form that can be compiled by the Absoft Fortran 77 compiler and run on a Power Macintosh personal computer. There are two input files, "crin" and "cring". "crin" is a user-supplied file which provides user preferences and parameters specific to the structure(s) in question. "cring" provides the probability distribution parameters detailed in Section 2.3. It is not intended to be modified by the user. There is one output file, "crou". The input and output files are described briefly after the code. These versions of the code and input files have not been validated. The author and Lawrence Livermore National Laboratory assume no legal liability or responsibility for their accuracy or use.

4.1. The Fortran code for Power Macintosh computation of $P_W(E)$.

```
implicit double precision (a-h,o-z)
common /phase/ indto
common /crits/ vcrit(5)
common /ncnc/ ncrit
common /probs/ oprob,rprob,pside,pend
common /prk/ prk(2,10,5,2,11,5)
common /loop/ nigloo,nrunway,ntype,ltype(11)
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /condc/ pcondo(11,2),qcondo(11,2)
write (*,*) 'here we go'
open (1,file="crin",status="old")
open (2,file="crou",status="unknown",action="write")
open (3,file="cring",status="old")
call initial
```



```

write (*,*) 'initialization is over'
do 80 i=1,nigloo
call sethr(i)
do 70 j=1,nrunway
call setrun(j)
do 60 k=1,2
call setloc(i,j,k)
do 50 l=1,ntype
ll=ltype(l)
call setype(i,j,k,ll)
call igloc(i,l,lok)
if (lok.eq.1) go to 20
write (2,10) i,j,k,ll
10 format(//,"target location is unreasonable for the following case:
1",//,"target",i3,//,"runway",i3,//,"direction",i2,//,"type",i2,/)
go to 50
20 do 40 m=1,ncrit
vc=vcrit(m)
vc2=vc*vc
do 30 indto=1,2
call adjust
call compute
prk(indto,i,j,k,l,m)=oprob*pcondo(l,indto)+rprob*qcondo(l,indto)
write (*,25) indto,i,j,k,ll,m,oprob,rprob,prk(indto,i,j,k,l,m)
25 format(6i3,3(1pe10.2))
30 continue
40 continue
50 continue
60 continue
70 continue
80 continue
call output
stop
end
subroutine initial
implicit double precision (a-h,o-z)
dimension runwid(5),gaskw(6),itype(11),nmilx(2),nmily(2),mj0(2)
common /loop/ nigloo,nrunway,ntype,ltype(11)
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /wyx/ nwyxt,nwyxl,wyxt(3),wyxtsh,wyxl(3),wyxlsh
common /wphiv/ wphiv(3),wphivsh,wphi0
common /wpwp/ nwpv
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
common /crits/ vcrit(5)
common /ncnc/ ncrit
common /runpr/ pruse(5),prto(5),prdir1(5),prmo(5,11)
common /runpar/ qrwlj(5),halfw(5),epshj(5),epshm(5),
1xtal(5),xtkj(5),clsidetj(5),clsidelj(5),clendj(5),c2endj(5)
common /igpar/ higlooi(10),riglooi(10),wpril(10,11),wpr2il(10,11),
1xigloc(10,5,2),yigloc(10,5)
common /acpar/ rac1(11)
common /crash/ pcrash(11,2)
common /crr/ noper
common /xytl/ xmint,xmaxt,ymaxt,xminl,xmaxl,ymaxl
common /xln/ xtmu,xtsig,xtal,xtk,xlmu,xlsig,xlal,xlk
common /yx/ yxz1,yxzlp,yxzlm,ymin,ymax,yxa,ypc,ypc1,yxk,yxkca
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
common /v0/ v0min,v0max,v0max2,v0a,v0c,v0k,v0k1

```

```

common /dcel/ dcelmin,dcelmax,dcelm,dcells,dcelk
common /rcel/ rcelmin,rcelmax,rcelm,rcels,rcelk
common /don/ aldon,bedon
common /betasg/ c1land, c2land,c3land,c1to,c2to,c3to
common /betase/ t1side, t2side,t3side,t1end,t2end,t3end
common /xsidetl/ c1sidet,c2sidet,c3sidet,c1sidel,c2sidel,c3sidel
common /yend/ c1end,c2end,c3end
common /eps/ epsilon,epsm,epsh,epshm
common /condc/ pcondo(11,2),qcondo(11,2)
common /flip/ iyneg(10,5),iflip
common /mill/ xmintm,ymintm,xminlm,yminlm,xmaxtm,ymaxtm,xmaxlm,
1 ymaxlm,flmil(16,16,2,2),cmil,ymilm(2),ymaxm(2)
common /genl/ xming,yming,xmaxg,ymaxg,flgen(20,20,5)
common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1 p5max(3,2),cp5(3,2)
common /pear6/ pear6a,pear6b,pear6c,p6mx,comp6,p6a(2),p6b(2),
1 p6c(2),p6max(2),cp6(2)
common /zlog/ zlmm,zlms,zlmc,zlgm,zlgs,zlgc,zlogm(3,2),
1 zlogsd(3,2),zlogc(2)
common /milphi/ clpm,c2pm,c3pm,clphim(2),c2phim(2),c3phim(2)
common /mskid/ skidm(2,2)
common /milth/ zmhm,zmhsd,zmhc,zmilhm(2,2),zmilhs(2,2),zmilhc(2,2)
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
common /gask/ gaskmu(6),gaskvk(5),gaskphi(5),gasksig
data epsilon,half /0.d0,0.5d0/
data cmil,cmil2,cfacm1,cfacm2,cfacm3 /5280.d0,27878400.d0,
11.0144092219d0,1.02923976608d0,1.01131812641d0/
data xmintm,ymintm,xmaxtm,ymaxtm /-45440.d0,-13200.d0,
112640.d0,55440.d0/
data xminlm,yminlm,xmaxlm,ymaxlm /-2640.d0,-29040.d0,
181840.d0,55440.d0/
data nmilx(1),nmilx(2),nmily(1),nmily(2) /11,16,13,16/
data mj0(1),mj0(2) /3,6/
data cgen2,cfacg /27878400.d0,1.01363467789d0/
data xming,xmaxg,ymaxg /-47800.d0,57800.d0,52800.d0/
data fkconv /1.687806/
data ntott /11/
epsm=-epsilon
pi=4.d0*datan(1.d0)
hpi=half*pi
hpil=log(hpi)
hpin=-hpi
pi2=pi+pi
spi=dsqrt(pi2)
pin=-pi
rpi=1.d0/pi
rpi2=1.d0/pi2
read (1,*) (itype(i),i=1,ntott)
ntype=0
do 10 l=1,ntott
read (3,*) wingt
if (itype(l).eq.0) go to 10
ntype=ntype+1
ltype(ntype)=1
racl(ntype)=half*wingt
10 continue
if (ntype.eq.0) stop
read (1,*) nigloo,nrunway

```

```

        if ((nigloo.gt.10).or.(nrunway.gt.5)) go to 400
        do 30 j=1,nrunway
        read (1,*) qrwlj(j),runwid(j)
        halfrwj(j)=half*runwid(j)
        epshj(j)=halfrwj(j)+epsilon
        epshmj(j)=-epshj(j)
        read (1,*) pruse(j),prto(j),prdir1(j)
        do 20 l=1,ntype
        read (1,*) prmo(j,l)
20      continue
30      continue
        do 70 i=1,nigloo
        read (1,*) higlooi(i),diam
        riglooi(i)=half*diam
        do 40 l=1,ntype
        wpril(i,l)=riglooi(i)+racl(l)
        wpr2il(i,l)=wpril(i,l)*wpril(i,l)
40      continue
        do 60 j=1,nrunway
        read (1,*) xigloo,yigloo
        if (yigloo.ge.0.d0) go to 45
        iyneg(i,j)=1
        yigloo=-yigloo
        go to 50
45      iyneg(i,j)=0
50      yigloc(i,j)=yigloo
        xigloc(i,j,2)=-xigloo
        xigloc(i,j,1)=xigloo+qrwlj(j)
60      continue
70      continue
        read (1,*) ncrit
        read (1,*) (vcrit(i),i=1,ncrit)
        read (3,*) pendd
        psidel=half*(1.d0-pendd)
        read (3,*) epsy
        read (3,*) xmint,xtmu,xtsig
        do 110 j=1,nrunway
        xmaxt=qrwlj(j)
        xtal=log(xmaxt-xmint)
        zaa=(xtal-xtmu)/xtsig
        call mdnor(zaa,zpp)
        xtk=1.d0/(spi*zpp*xtsig)
        xtalj(j)=xtal
        xtkj(j)=xtk
110     continue
        read (3,*) nwyxt
        read (3,*) (wyxt(i),i=1,nwyxt)
        read (3,*) wyxtsh
        ymxt=(-log(epsy))**(1.d0/wyxtsh)
        read (3,*) xminl,xmaxl,xlmux,xlsig
        xlal=log(xmaxl-xminl)
        zaa=(xlal-xlmux)/xlsig
        call mdnor(zaa,zpp)
        xlk=1.d0/(spi*zpp*xlsig)
        read (3,*) nwyxl
        read (3,*) (wyxl(i),i=1,nwyxl)
        read (3,*) wyxlsh
        ymaxl=(-log(epsy))**(1.d0/wyxlsh)

```

```

      read (3,*) asidet,bsidet,asidel,bsidel
      gcont=bgama(asidet,bsidet)
      c2sidet=asidet-1.d0
      c3sidet=bsidet-1.d0
      gconl=bgama(asidel,bsidel)
      c2sidel=asidel-1.d0
      c3sidel=bsidel-1.d0
      do 140 j=1,nrunway
      clsidetj(j)=gcont/grwlj(j)
      clsidelj(j)=gconl/grwlj(j)
140    continue
      read (3,*) c0
      c02=c0+c0
      cconl=bgama(c0,c0)
      do 150 j=1,nrunway
      clendj(j)=cconl/(runwid(j)**(c02-1.d0))
      c2endj(j)=halfwrj(j)*halfwrj(j)
150    continue
      c3end=c0-1.d0
      read (3,*) vkmin,vkmax,vka,vkc
      vkcl=vkc-1.d0
      vkk=1.d0/(1.d0-dexp(-(vkmax-vkmin)/vka)**vkc))
      vkk1=1.d0-vkk
      vkkca=vkk*(vkc/vka)
      vkmax2=vkmax*vkmax
      read (3,*) v0min,v0max,v0a,v0c
      v0k=1.d0/(1.d0-dexp(-(v0max-v0min)/v0a)**v0c))
      v0k1=1.d0-v0k
      v0max2=v0max*v0max
      read (3,*) dcelmin,dcelmax,dcelm,dcells
      zau=(log(dcelmax)-dcelm)/dcells
      zal=(log(dcelmin)-dcelm)/dcells
      call mdnor(zau,zpu)
      call mdnor(zal,zpl)
      dcelk=1.d0/(spi*dcells*(zpu-zpl))
      read (3,*) rcelmin,rcelmax,rcelm,rcels
      zau=(log(rcelmax)-rcelm)/rcels
      zal=(log(rcelmin)-rcelm)/rcels
      call mdnor(zau,zpu)
      call mdnor(zal,zpl)
      rcelk=1.d0/(spi*rcels*(zpu-zpl))
      read (3,*) nwpv
      read (3,*) (wphiv(i),i=1,nwpv)
      read (3,*) wphivsh
      read (3,*) wphi0
      wphi0=wphi0*pi/180.d0
      read (3,*) aldon,bedon
      read (3,*) ato,bto
      clto=rpi2*bgama(ato,bto)
      c2to=ato-1.d0
      c3to=bto-1.d0
      read (3,*) atl,btl
      clland=rpi2*bgama(atl,btl)
      c2land=atl-1.d0
      c3land=btl-1.d0
      read (3,*) aside,bside
      t1side=rpi*bgama(aside,bside)
      t2side=aside-1.d0

```

```

t3side=bside-1.d0
read (3,*) aend,bend
t1end=rpi*bgama(aend,bend)
t2end=aend-1.d0
t3end=bend-1.d0
do 220 icat=1,2
do 210 k=1,2
z=0.d0
ik=nmilx(k)
jk=nmily(k)
do 170 i=1,ik
read (3,*) (flmil(i,j,icat,k),j=1,jk)
do 160 j=1,jk
z=z+flmil(i,j,icat,k)
160 continue
170 continue
aden=z*cmil2
do 190 i=1,ik
do 180 j=1,jk
flmil(i,j,icat,k)=flmil(i,j,icat,k)/aden
180 continue
190 continue
if (k.eq.2) go to 200
i0=ik
j0=mj0(1)
flmil(i0,j0,icat,1)=flmil(i0,j0,icat,1)*cfacm1
i0=i0-1
flmil(i0,j0,icat,1)=flmil(i0,j0,icat,1)*cfacm2
i0=i0-1
flmil(i0,j0,icat,1)=flmil(i0,j0,icat,1)*cfacm3
go to 210
200 j0=mj0(2)
flmil(1,j0,icat,2)=flmil(1,j0,icat,2)*cfacm1
flmil(2,j0,icat,2)=flmil(2,j0,icat,2)*cfacm2
flmil(3,j0,icat,2)=flmil(3,j0,icat,2)*cfacm3
210 continue
220 continue
yming=-ymaxg
do 270 icat=1,5
z=0.d0
do 240 j=1,20
read (3,*) (flgen(i,j,icat),i=1,20)
do 230 i=1,20
z=z+flgen(i,j,icat)
230 continue
240 continue
aden=z*cgen2
do 260 i=1,20
do 250 j=1,20
flgen(i,j,icat)=flgen(i,j,icat)/aden
250 continue
260 continue
flgen(10,10,icat)=flgen(10,10,icat)*cfacg
flgen(10,11,icat)=flgen(10,11,icat)*cfacg
flgen(11,10,icat)=flgen(11,10,icat)*cfacg
flgen(11,11,icat)=flgen(11,11,icat)*cfacg
270 continue
do 290 i=1,3

```

```

do 280 j=1,2
  read (3,*) p5a(i,j),p5b(i,j),p5max(i,j)
  cp5(i,j)=p5b(i,j)*dexp(algama(p5a(i,j)))
280  continue
290  continue
  do 300 j=1,2
    read (3,*) p6a(j),p6b(j),p6c(j),p6max(j)
    cp6(j)=p6c(j)/bgama(p6a(j),p6b(j))
300  continue
    do 320 i=1,3
      do 310 j=1,2
        read (3,*) zlogm(i,j),zlogsd(i,j)
310  continue
320  continue
        do 330 j=1,2
          z=(hpil-zlogm(3,j))/zlogsd(3,j)
          call mdnor(z,zlogc(j))
330  continue
          read (3,*) aphil,bphil,aphis,bphis
          gconl=bgama(aphil,bphil)
          c1phim(1)=gconl/hpi
          c2phim(1)=aphil-1.d0
          c3phim(1)=bphil-1.d0
          gcons=bgama(aphis,bphis)
          c1phim(2)=gcons/hpi
          c2phim(2)=aphis-1.d0
          c3phim(2)=bphis-1.d0
          read (3,*) skidm(1,1),skidm(1,2),skidm(2,1),skidm(2,2)
          do 350 i=1,2
            do 340 j=1,2
              read (3,*) zmilhm(i,j),zmilhs(i,j)
              z1=(pi-zmilhm(i,j))/zmilhs(i,j)
              z2=(-pi-zmilhm(i,j))/zmilhs(i,j)
              call mdnor(z1,zz1)
              call mdnor(z2,zz2)
              zmilhc(i,j)=(zz1-zz2)*spi*zmilhs(i,j)
340  continue
350  continue
              do 360 i=1,5
                read (3,*) zp,zpa
                gaphi(i)=pi/zp
                gaphia(i)=zpa/pi
360  continue
                do 370 i=1,5
                  read (3,*) zv,zvb
                  gav(i)=zv*fkconv
                  gavb(i)=zvb/gav(1)
370  continue
                  read (3,*) gaskb,gaskm,gasksig
                  read (3,*) (gaskw(i),i=1,6)
                  read (3,*) (gaskvk(i),i=1,5)
                  read (3,*) (gaskphi(i),i=1,5)
                  do 380 i=1,6
                    gaskmu(i)=gaskb+gaskm*log(gaskw(i))
380  continue
                  read (1,*) noper
                  j=0
                  do 390 l=1,ntott

```

```

      read (3,*) pcrt,pcrl,pcont,pconl
      if (itype(1).eq.0) go to 390
      j=j+1
      pcrash(j,1)=pcrt
      pcrash(j,2)=pcrl
      pcondo(j,1)=pcont
      pcondo(j,2)=pconl
      qcondo(j,1)=1.d0-pcont
      qcondo(j,2)=1.d0-pconl
390  continue
      close (1)
      close (3)
      call wrt
      return
400  write (*,410)
410  format(/,"program cannot handle more than 10 targets or 5 runways"
1,/)
      stop
      end
      subroutine sethr(i)
      implicit double precision (a-h,o-z)
      common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
      common /igpar/ higlooi(10),riglooi(10),wpril(10,11),wpr2il(10,11),
1xigloc(10,5,2),yigloc(10,5)
      higloo=higlooi(i)
      rigloo=riglooi(i)
      return
      end
      subroutine setrun(j)
      implicit double precision (a-h,o-z)
      common /xsidet1/ clsidet,c2sidet,c3sidet,clsidel,c2sidel,c3sidel
      common /yend/ clend,c2end,c3end
      common /eps/ epsilon,epsm,epsh,epshm
      common /xytl/ xmint,xmaxt,ymaxt,xminl,xmaxl,ymaxl
      common /xln/ xtmu,xtsig,xtal,xtk,xlmu,xlsig,xlal,xlk
      common /runpar/ qrwlj(5),halfrwj(5),epshj(5),epshmj(5),
1xtalj(5),xtkj(5),clsidetj(5),clsidelj(5),clendj(5),c2endj(5)
      common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
      qrw1=qrwlj(j)
      xmaxt=qrw1
      halfrw=halfrwj(j)
      epsh=epshj(j)
      epshm=epshmj(j)
      xtal=xtalj(j)
      xtk=xtkj(j)
      clsidet=clsidetj(j)
      clsidel=clsidelj(j)
      clend=clendj(j)
      c2end=c2endj(j)
      return
      end
      subroutine setloc(i,j,k)
      implicit double precision (a-h,o-z)
      common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
      common /igpar/ higlooi(10),riglooi(10),wpril(10,11),wpr2il(10,11),

```

```

1xigloc(10,5,2),yigloc(10,5)
  xigloo=xigloc(i,j,k)
  yigloo=yigloc(i,j)
  return
end
subroutine setype(i,j,k,l)
  implicit double precision (a-h,o-z)
  common /mill/ xmintm,ymintm,xminlm,yminlm,xmaxtm,ymaxtm,xmaxlm,
1ymaxlm,flmil(16,16,2,2),cmil,ymirm(2),ymaxm(2)
  common /codes/ itcode,imilc,igact,igacl,igac,igas
  common /flip/ iyneg(10,5),iflip
  common /flyin5/ fip5a,fip5b,fip5mx,fip5c
  common /genl/ xming,yming,xmaxg,ymaxg,flgen(20,20,5)
  common /pars/ xmin,xmax,psidel,pendd,vc,vc2
  common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1p5max(3,2),cp5(3,2)
  common /pear6/ pear6a,pear6b,pear6c,p6mx,comp6,p6a(2),p6b(2),
1p6c(2),p6max(2),cp6(2)
  common /zlog/ zlmm,zlms,zlmc,zlgm,zlgs,zlgc,zlogm(3,2),
1zlogsd(3,2),zlogc(2)
  common /milphi/ c1pm,c2pm,c3pm,c1phim(2),c2phim(2),c3phim(2)
  common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
  if ((1.lt.3).or.(1.eq.9)) go to 10
  iflip=0
  if (1.gt.4) go to 20
  itcode=2
  imilc=1-2
  pear6a=p6a(imilc)-1.d0
  pear6b=p6a(imilc)+p6b(imilc)
  pear6c=p6c(imilc)
  comp6=cp6(imilc)
  p6mx=p6max(imilc)
  fip5a=p5a(3,imilc)+1.d0
  fip5b=p5b(3,imilc)
  fip5c=cp5(3,imilc)
  fip5mx=p5max(3,imilc)
  zlgm=zlogm(3,imilc)
  zlgs=zlogsd(3,imilc)
  zlgc=zlogc(imilc)
  c1pm=c1phim(imilc)
  c2pm=c2phim(imilc)
  c3pm=c3phim(imilc)
  if (iyneg(i,j).eq.1) go to 5
  yminm(1)=ymintm
  yminm(2)=yminlm
  ymaxm(1)=ymaxtm
  ymaxm(2)=ymaxlm
  return
5  iflip=1
  yminm(1)=-ymaxtm
  yminm(2)=-ymaxlm
  ymaxm(1)=-ymintm
  ymaxm(2)=-yminlm
  return
10  itcode=1
  return
20  itcode=3
  igas=1-5

```



```

        if (igas.eq.0) igas=6
        xmin=xming
        xmax=xmaxg
        ymin=yming
        ymax=ymaxg
        if ((1.eq.7).or.(1.eq.8)) go to 30
        igact=1
        igacl=2
25      k2=2-k
        if (iyNEG(i,j).eq.k2) iflip=1
        return
30      igact=3
        igacl=1-3
        go to 25
    end
    subroutine igloc(i,l,lok)
    implicit double precision (a-h,o-z)
    common /igpar/ higlooi(10),riglooi(10),wpril(10,11),wpr2il(10,11),
1xigloc(10,5,2),yigloc(10,5)
    common /acpar/ rac(11)
    common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
    rac=rac(1)
    wpr=wpril(i,1)
    wpr2=wpr2il(i,1)
    xtmb=xigloo-wpr
    xtpb=xigloo+wpr
    ytmb=yigloo-wpr
    call toocl(lok)
    return
    end
    subroutine toocl(lok)
    implicit double precision (a-h,o-z)
    common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
    lok=1
    if (ytmb.ge.halfrw) return
    if (yigloo.lt.0.d0) go to 60
    if (yigloo.gt.halfrw) go to 10
    if ((xtpb.gt.0.d0).and.(xtmb.lt.qrw1)) go to 40
    return
10    yh=yigloo-halfrw
    wy=wpr2-yh*yh
    if (xigloo.lt.0.d0) go to 20
    xq=xigloo-qrw1
    if (xq) 40,40,30
20    if (xigloo*xigloo.lt.wy) go to 40
    return
30    if (xq*xq.lt.wy) go to 40
    return
40    write (*,50)
50    format(//,"igloo located unreasonably close to runway",/)
    lok=0
    return
60    write (*,70)
70    format(//,"igloo cannot have negative y-coordinate",/)
    lok=0
    return

```

```

end
subroutine wrt
implicit double precision (a-h,o-z)
common /loop/ nigloo,nrunway,ntype,ltype(11)
common /crash/ pcrash(11,2)
common /crrc/ noper
common /runpr/ pruse(5),prto(5),prdir1(5),prmo(5,11)
common /igpar/ higlooi(10),riglooi(10),wpril(10,11),wpr2il(10,11),
1xigloc(10,5,2),yigloc(10,5)
common /acpar/ racl(11)
common /runpar/ qrwlj(5),halfwrj(5),epshj(5),epshmj(5),
1xtalj(5),xtkj(5),clsidetj(5),clsidelj(5),clendj(5),c2endj(5)
common /flip/ iyneg(10,5),iflip
data two /2.d0/
5   format("jet crash study",30x,////)
   write (2,10) noper
10  format("number of operations in time window:",i6,///,
1   "crash probabilities",///,"type t/o   lan")
   do 20 l=1,ntype
   write (2,15) ltype(l),pcrash(l,1),pcrash(l,2)
15  format(i3,2(1pe8.1))
20  continue
   write (2,25)
25  format(///)
   do 35 l=1,ntype
   write (2,30) ltype(l),two*racl(l)
30  format("type:",i2,5x,"wingspan:",f6.0)
35  continue
   write (2,40) nigloo,nrunway
40  format(///,"number of targets:",i3,/, "number of runways:",i3,///,
1   "target height diameter")
   do 60 i=1,nigloo
   write (2,50) i,higlooi(i),two*riglooi(i)
50  format(i4,5x,f4.0,4x,f5.0)
60  continue
   write (2,70)
70  format(///,"runway length width puse pt/o pdr1")
   do 80 j=1,nrunway
   write (2,75) j,qrwlj(j),two*halfwrj(j),pruse(j),prto(j),prdir1(j)
75  format(i4,4x,f6.0,f7.0,3f6.2)
80  continue
   do 95 j=1,nrunway
   write (2,85)
85  format(///,"runway type freqs")
   write (2,90) j,(prmo(j,l), l=1,ntype)
90  format(i4,3x,10f6.2)
95  continue
   write (2,100)
100 format(///,"target locations (2 directions) by runway")
   do 140 i=1,nigloo
   write (2,110) i
110 format(//,"target",i3)
   do 130 j=1,nrunway
   yig=yigloc(i,j)
   if (iyneg(i,j).eq.1) yig=-yig
   write (2,120) j,xigloc(i,j,1)-qrwlj(j),yig,
1xigloc(i,j,2)-qrwlj(j),yig
120 format(i2,2(" (",f7.0," ",f7.0,")"))

```

```

130  continue
140  continue
    return
    end
    subroutine adjust
    implicit double precision (a-h,o-z)
    common /phase/ indto
    common /xytl/ xmint,xmaxt,ymaxt,xminl,xmaxl,ymaxl
    common /wyx/ nwyxt,nwyxl,wyxt(3),wyxtsh,wyxl(3),wyxlsh
    common /xsidetl/ c1sidet,c2sidet,c3sidet,c1sidel,c2sidel,c3sidel
    common /pars/ xmin,xmax,psidel,pendd,vc,vc2
    common /yx/ yxz1,yxzlp,yxzlm,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
    common /xside/ clside,c2side,c3side
    common /mill/ xmintm,ymintm,xminlm,yminlm,xmaxtm,ymaxtm,xmaxlm,
1ymaxlm,flmil(16,16,2,2),cmil,yminm(2),ymaxm(2)
    common /codes/ itcode,imilc,igact,igac1,igac,igas
    common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1p5max(3,2),cp5(3,2)
    common /zlog/ zlmm,zlms,zlmc,zlgm,zlgs,zlgc,zlogm(3,2),
1zlogsd(3,2),zlogc(2)
    common /milth/ zmhm,zmhsd,zmhc,zmilhm(2,2),zmilhs(2,2),zmilhc(2,2)
    common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
    if (indto.eq.1) go to 20
    clside=c1sidel
    c2side=c2sidel
    c3side=c3sidel
    if (itcode-2) 5,10,15
5    xmin=xminl
    xmax=xmaxl
    yxc=wyxlsh
    yxc1=yxc-1.d0
    return
10   xmin=xminlm
    xmax=xmaxlm
12   ymin=yminm(indto)
    ymax=ymaxm(indto)
    zmhm=zmilhm(indto,imilc)
    zmhsd=zmilhs(indto,imilc)
    zmhc=zmilhc(indto,imilc)
    pear5a=p5a(indto,imilc)+1.d0
    pear5b=p5b(indto,imilc)
    comp5=cp5(indto,imilc)
    p5mx=p5max(indto,imilc)
    zlmm=zlogm(indto,imilc)
    zlms=zlogsd(indto,imilc)
    zlmc=spi*zlms
    return
15   igac=igac1
    return
20   clside=c1sidet
    c2side=c2sidet
    c3side=c3sidet
    if (itcode-2) 25,30,35
25   xmin=xmint
    xmax=xmaxt
    yxc=wyxtsh
    yxc1=yxc-1.d0
    return

```

```

30  xmin=xmintm
    xmax=xmaxtm
    go to 12
35  igac=igact
    return
    end
    subroutine compute
    implicit double precision (a-h,o-z)
    common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrwl,halfrw,wpr,wpr2,
1higloo,rigloo,rac
    common /pars/ xmin,xmax,psidel,pendd,vc,vc2
    common /probs/ oprob,rprob,pside,pend
    common /yx/ yxz1,yxzlp,yxzlm,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
    common /pvkc/ pvkc
    common /codes/ itcode,imilc,igact,igac1,igac,igas
    external g1,g2a,g2b,g2c,gside,gend
    pvkc=1.d0
    if (vc.gt.0.d0) pvkc=pdirect(vc)
    oprob=0.d0
    if (pvkc.eq.0.d0) go to 50
    if (xtpb.gt.xmin) go to 10
    x1=xmin
    go to 40
10  if (xtmb.ge.xmin) go to 20
    x1=xmin
    x2=xtpb
    go to 30
20  x1=xmin
    x2=xtmb
    if (xtmb.gt.xmax) x2=xmax
    oprob=ggaus1(g1,x1,x2)
    if (xtmb.ge.xmax) go to 50
    x1=xtmb
    x2=xtpb
    if (xtpb.gt.xmax) x2=xmax
30  oprob=oprob+ggaus1(g2a,x1,x2)
    ymaxx=ymax
    if (itcode.gt.1) go to 35
    call prepx(x1)
    ymax1=ymax
    call prepx(x2)
    ymaxx=min(ymax,ymax1)
35  if (yigloo.lt.ymaxx) oprob=oprob+ggaus1(g2b,x1,x2)
    if (ytmb.lt.ymaxx) oprob=oprob+pvkc*ggaus1(g2c,x1,x2)
    if (xtpb.ge.xmax) go to 50
    x1=xtpb
40  x2=xmax
    oprob=oprob+ggaus1(g1,x1,x2)
50  pv0c=fccdfv0(vc)
    pside=0.d0
    pend=0.d0
    if (pv0c.eq.0.d0) go to 60
    pside=psidel*ggaus1(gside,0.d0,qrwl)
    if (xtmb.ge.0.d0) go to 60
    call prepx(0.d0)
    pend=pendd*ggaus1(gend,-halfrw,halfrw)
60  rprob=pside+pend
    return

```

```

end
double precision function pdirect(vc)
implicit double precision (a-h,o-z)
common /phase/ indto
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
common /codes/ itcode,imilc,igact,igacl,igac,igas
common /flyin5/ fip5a,fip5b,fip5mx,fip5c
common /pear6/ pear6a,pear6b,pear6c,p6mx,comp6,p6a(2),p6b(2),
1p6c(2),p6max(2),cp6(2)
external fip5,fp6
if (itcode-2) 10,20,60
10 pdirect=fccdfvk(vc)
return
20 if (indto.eq.1) go to 40
if (vc.lt.fip5mx) go to 30
pdirect=0.d0
return
30 pdirect=1.d0-ggaus1(fip5,0.d0,vc)
return
40 if (vc.lt.p6mx) go to 50
pdirect=0.d0
return
50 pdirect=1.d0-ggaus1(fp6,0.d0,vc)
return
60 if (vc.lt.gav(5)) go to 70
pdirect=0.d0
return
70 z=0.d0
w=0.d0
do 90 i=1,5
if (vc.lt.gav(i)) go to 80
z=z+(gav(i)-w)*gavb(i)
w=gav(i)
go to 90
80 z=z+(vc-w)*gavb(i)
go to 100
90 continue
100 pdirect=1.d0-z
return
end
subroutine output
implicit double precision (a-h,o-z)
dimension tble(11,6,5,2),alpha(6)
common /crash/ pcrash(11,2)
common /crcr/ noper
common /runpr/ pruse(5),prto(5),prdir1(5),prmo(5,11)
common /prk/ prk(2,10,5,2,11,5)
common /loop/ nigloo,nrunway,ntype,ltype(11)
common /crits/ vcrit(5)
common /ncnc/ ncrit
data alpha /" 1 "," 2 "," 3 "," 4 "," 5 "," 6 "/
data any /"any"/
zn=noper
nr1=nrunway+1
alpha(nr1)=any
nr1=nigloo+1
do 30 i=1,nigloo
do 20 m=1,ncrit

```

```

sum=0.d0
do 10 j=1,nrunway
  pit=prto(j)
  pil=1.d0-pit
  zl1=prdir1(j)
  zl2=1.d0-zl1
  b1=pit*zl1
  b2=pit*zl2
  b3=pil*zl1
  b4=pil*zl2
  c=zn*pruse(j)
  z1=0.d0
  z2=0.d0
  z3=0.d0
  z4=0.d0
  do 5 l=1,ntype
    a=prmo(j,l)*pcrash(1,1)
    b=prmo(j,l)*pcrash(1,2)
    z1=z1+a*prk(1,i,j,1,1,m)
    z2=z2+a*prk(1,i,j,2,1,m)
    z3=z3+b*prk(2,i,j,1,1,m)
    z4=z4+b*prk(2,i,j,2,1,m)
5  continue
    z=b1*z1+b2*z2+b3*z3+b4*z4
    tble(i,j,m,1)=z
    cz=c*z
    tble(i,j,m,2)=cz
    sum=sum+cz
10  continue
    tble(i,nr1,m,2)=sum
20  continue
30  continue
    do 60 m=1,ncrit
      gtot=0.d0
      do 50 j=1,nrunway
        tot=0.d0
        do 40 i=1,nigloo
          tot=tot+tble(i,j,m,2)
40  continue
          tble(ni1,j,m,2)=tot
          gtot=gtot+tot
50  continue
          tble(ni1,nr1,m,2)=gtot
60  continue
          write (2,70)
70  format(////////,"rates of hitting target with velocity exceeding",
1" vc for (runway,target) pairs")
          do 110 m=1,ncrit
            write (2,80) vcrit(m),(j,j=1,nrunway)
80  format(//,"vc:",f5.0,/,5i10)
            do 100 i=1,nigloo
              write (2,90) i,(tble(i,j,m,1),j=1,nrunway)
90  format(i2,1x,6(1pe10.2))
100 continue
110 continue
          write (2,120)
120 format(////////,"probabilities of hitting target with velocity",
1" exceeding vc for (runway,target) pairs")

```

```

do 150 m=1,ncrit
write (2,125) vcrit(m),(alpha(j),j=1,nr1)
125 format(//,"vc:",f5.0,/,1x,6(7x,a3))
do 130 i=1,nigloo
write (2,90) i,(tble(i,j,m,2),j=1,nr1)
130 continue
write (2,140) any,(tble(ni1,j,m,2),j=1,nr1)
140 format(a3,6(1pe10.2))
150 continue
close (2)
return
end
subroutine prepx(x)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /xx/ x0,xdabs,xd2,indx
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
lhigloo,rigloo,rac
common /phase/ indto
common /wyx/ nwyxt,nwyxl,wyxt(3),wyxtsh,wyxl(3),wyxlsh
common /yx/ yxz1,yxzlp,yxzlm,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
common /eps/ epsilon,epsm,epsh,epshm
common /xyt1/ xmint,xmaxt,ymaxt,xmin1,xmax1,ymax1
x0=x
xdabs=abs(x-xigloo)
xd2=xdabs*xdabs
indx=1
if (x.lt.xigloo) indx=0
if ((x.ge.0.d0).and.(x.le.qrw1)) go to 10
yxz1=0.d0
yxzlp=epsilon
yxzlm=epsm
go to 20
10 yxz1=halfrw
yxzlp=epsh
yxzlm=epshm
20 if (itcode.gt.1) return
if (indto.eq.1) go to 30
yxa=dexp(weibp(x,nwyxl,wyxl))
ymax=ymax1*yxa
ymin=-ymax
yxk=0.5d0/(1.d0-dexp(-(ymax-yxz1)/yxa)**yxc))
yxkca=yxk*(yxc/yxa)
return
30 yxa=dexp(weibp(x,nwyxt,wyxt))
ymax=ymaxt*yxa
ymin=-ymax
yxk=0.5d0/(1.d0-dexp(-(ymax-yxz1)/yxa)**yxc))
yxkca=yxk*(yxc/yxa)
return
end
subroutine prepy(y)
implicit double precision (a-h,o-z)
common /yy/ y0,alpha,beta,iout
common /xx/ x0,xdabs,xd2,indx
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
lhigloo,rigloo,rac
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu

```

```

y0=y
iout=1
if (y.lt.0.d0) go to 10
if (y.le.yigloo) iout=0
10 yd=y-yigloo
   yd2=yd*yd
   d2=xd2+yd2
   delam2=wpr2/d2
   alpha=acus(xdabs/dsqrt(d2))
   beta=acus(dsqrt(1.d0-delam2))
   return
end
double precision function fpdfx(x)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /phase/ indto
common /xytl/ xminl,xmaxt,ymaxt,xminl,xmaxl,ymaxl
common /xln/ xtmu,xtsig,xtal,xtk,xlmu,xlsig,xlal,xlk
if (itcode.eq.1) go to 5
fpdfx=1.d0
return
5  if (indto.eq.1) go to 10
   z=x-xminl
   ze=(log(z)-xlmu)/xlsig
   fpdfx=xlk/(z*dexp(0.5d0*ze*ze))
   return
10 z=xmaxt-x
   ze=(log(z)-xtmu)/xtsig
   fpdfx=xtk/(z*dexp(0.5d0*ze*ze))
   return
end
double precision function fccdfvk(v)
implicit double precision (a-h,o-z)
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
if (v.ge.vkmax) go to 10
if (v.le.vkmin) go to 20
fccdfvk=vkk1+vkk*dexp(-(v-vkmin)/vka)**vkc)
return
10 fccdfvk=0.d0
   return
20 fccdfvk=1.d0
   return
end
double precision function fpdfvk(v)
implicit double precision (a-h,o-z)
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
z=(v-vkmin)/vka
fpdfvk=vkkca*(z**vkc1)*dexp(-z**vkc)
return
end
double precision function fccdfv0(v)
implicit double precision (a-h,o-z)
common /v0/ v0min,v0max,v0max2,v0a,v0c,v0k,v0k1
if (v.ge.v0max) go to 10
if (v.le.v0min) go to 20
fccdfv0=v0k1+v0k*dexp(-(v-v0min)/v0a)**v0c)
return
10 fccdfv0=0.d0

```



```

return
20 fccdfv0=1.d0
return
end
double precision function fpdfdccl(a)
implicit double precision (a-h,o-z)
common /dcel/ dcelmin,dcelmax,dcelm,dcelr,dcelk
z=(log(a)-dcelm)/dcelr
fpdfdccl=dcelk*dexp(-0.5d0*z*z)/a
return
end
double precision function fpdfrccl(a)
implicit double precision (a-h,o-z)
common /rcel/ rcelmin,rcelmax,rcelm,rcelr,rcelk
z=(log(a)-rcelm)/rcelr
fpdfrccl=rcelk*dexp(-0.5d0*z*z)/a
return
end
double precision function g2a(x)
implicit double precision (a-h,o-z)
common /xx/ x0,xdabs,xd2,indx
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
external g0
call prepx(x)
y2=yigloo-dsqrt(wpr2-xd2)
if (y2.gt.ymax) go to 10
if (y2.gt.yxz1p) go to 20
if (y2.gt.yxz1m) y2=yxz1m
g2a=fpdfx(x)*ggaus2(g0,ymin,y2)
return
10 y2=ymax
20 a1=ggaus2(g0,ymin,yxz1m)
a2=ggaus2(g0,yxz1p,y2)
g2a=fpdfx(x)*(a1+a2)
return
end
double precision function g2b(x)
implicit double precision (a-h,o-z)
common /xx/ x0,xdabs,xd2,indx
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
external g0
call prepx(x)
y1=yigloo+dsqrt(wpr2-xd2)
if (y1.ge.ymax) go to 10
if (y1.lt.yxz1p) y1=yxz1p
g2b=fpdfx(x)*ggaus2(g0,y1,ymax)
return
10 g2b=0.d0
return
end
double precision function g2c(x)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,

```

```

1higloo,rigloo, rac
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,ycx,ycx1,yxk,yxkca
external fpdfyx,fpdfml,fpdfgl
call prepx(x)
xa=x-xigloo
sq=dsqrt(wpr2-xa*xa)
y1=yigloo-sq
y2=yigloo+sq
if (y1.lt.yxz1m) go to 40
if (y1.lt.yxz1p) y1=yxz1p
if (itcode-2) 10,20,30
10 g2c=fpdfx(x)*ggaus2(fpdfyx,y1,y2)
return
20 g2c=ggaus2(fpdfml,y1,y2)
return
30 g2c=ggaus2(fpdfgl,y1,y2)
return
40 if (itcode-2) 50,60,70
50 a1=ggaus2(fpdfyx,y1,yxz1m)
a2=ggaus2(fpdfyx,yxz1p,y2)
g2c=fpdfx(x)*(a1+a2)
return
60 a1=ggaus2(fpdfml,y1,yxz1m)
a2=ggaus2(fpdfml,yxz1p,y2)
g2c=a1+a2
return
70 a1=ggaus2(fpdfgl,y1,yxz1m)
a2=ggaus2(fpdfgl,yxz1p,y2)
g2c=a1+a2
return
end
double precision function g1(x)
implicit double precision (a-h,o-z)
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,ycx,ycx1,yxk,yxkca
external g0
call prepx(x)
a1=ggaus2(g0,ymin,yxz1m)
a2=ggaus2(g0,yxz1p,ymax)
g1=fpdfx(x)*(a1+a2)
return
end
double precision function g0(y)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /flip/ iyneg(10,5),iflip
common /mill/ xmintm,ymintm,xminlm,yminlm,xmaxtm,ymaxtm,xmaxlm,
lymaxlm,flmil(16,16,2,2),cmil,yminm(2),ymaxm(2)
common /genl/ xming,yming,xmaxg,ymaxg,flgen(20,20,5)
common /xx/ x0,xdabs,xd2,indx
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /phase/ indto
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,ycx,ycx1,yxk,yxkca
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
external f8,f9
if (itcode-2) 10,20,40
10 yf0=fpdfyx(y)
go to 60
20 ii=(x0-xmin)/cmil+1.d0

```

```

        if (iflip.eq.1) go to 25
        jj=(y-ymin)/cmil+1.d0
        go to 30
25      jj=(ymax-y)/cmil+1.d0
30      yf0=flmil(ii,jj,imilc,indto)
        go to 60
40      ii=(x0-xmin)/5280.d0+1.d0
        if (iflip.eq.1) go to 45
        jj=(y-ymin)/5280.d0+1.d0
        go to 50
45      jj=(ymax-y)/5280.d0+1.d0
50      yf0=flgen(ii,jj,igac)
60      if (yf0.le.0.d0) go to 90
        call prepy(y)
        call prept(1,th1,th2,th3,th4,ind2)
        zmu=thmul
        a1=ggaus3(f8,th1,th2)
        if (ind2.eq.0) go to 70
        zmu=thmu2
        a1=a1+ggaus3(f8,th3,th4)
70      call prept(0,th1,th2,th3,th4,ind2)
        zmu=thmul
        a2=ggaus3(f9,th1,th2)
        if (ind2.eq.0) go to 80
        zmu=thmu2
        a2=a2+ggaus3(f9,th3,th4)
80      g0=yf0*(a1+a2)
        return
90      g0=0.d0
        return
end
double precision function fpdfyx(y)
implicit double precision (a-h,o-z)
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,yxc,yxc1,yxk,yxkca
wa=(abs(y)-yxz1)/yxa
fpdfyx=yxkca*(wa**yxc1)/dexp(wa**yxc)
return
end
double precision function f1(v)
implicit double precision (a-h,o-z)
common /vlow/ vlow
f1=fpdfvk(v)*fcdfpv(acus(vlow/v),v)
return
end
double precision function f2(a)
implicit double precision (a-h,o-z)
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
common /vlow/ vlow
external f1
vlow=fvlow(a)
vll=max(vlow,vkmin)
f2=fpdfdc1(a)*ggaus5(f1,vll,vkmax)
return
end
double precision function f3()
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /phase/ indto

```

```

common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1p5max(3,2),cp5(3,2)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
common /dcel/ dcelmin,dcelmax,dcelm,dcelm,dcelk
common /mskid/ skidm(2,2)
external f2,f12,f13
if (itcode-2) 5,15,25
5 ahi=(vkmax2-vc2)/de2
if (ahi.gt.dcelmin) go to 10
f3=0.d0
return
10 if (ahi.gt.dcelmax) ahi=dcelmax
f3=ggaus4(f2,dcelmin,ahi)
return
15 ahi=(p5mx*p5mx-vc2)/de2
if (ahi.gt.0.d0) go to 20
f3=0.d0
return
20 f3=skidm(indto,imilc)*ggaus4(f12,0.d0,ahi)
return
25 f3=ggaus4(f13,0.d0,hpi)
return
end
double precision function f4()
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /v0/ v0min,v0max,v0max2,v0a,v0c,v0k,v0k1
common /rcel/ rcelmin,rcelmax,rcelm,rcelm,rcels,rcelk
external f5
ahi=(v0max2-vc2)/de2
if (ahi.gt.rcelmin) go to 10
f4=0.d0
return
10 if (ahi.gt.rcelmax) ahi=rcelmax
f4=ggaus3(f5,rcelmin,ahi)
return
end
double precision function f5(a)
implicit double precision (a-h,o-z)
common /v0/ v0min,v0max,v0max2,v0a,v0c,v0k,v0k1
v1=fvlow(a)
if (v1.gt.v0min) go to 10
f5=fpdfrcel(a)
return
10 f5=fpdfrcel(a)*fccdfv0(v1)
return
end
double precision function f6(v)
implicit double precision (a-h,o-z)
common /whig/ whig
common /wphiv/ wphiv(3),wphivsh,wphi0
common /wppv/ wppv
f6=fpdfvk(v)*dexp(-(whig/dexp(weibp(v,wppv,wphiv))))**wphivsh)
return

```

```

end
double precision function f7()
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igas
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /whig/ whig
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
common /wphiv/ wphiv(3),wphivsh,wphi0
common /vk/ vkmin,vkmax,vkmax2,vka,vkc,vkc1,vkk,vkk1,vkkca
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /phase/ indto
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
common /zlog/ zlmm,zlms,zlmc,zlgm,zlgs,zlgc,zlogm(3,2),
1zlogsd(3,2),zlogc(2)
common /pvkc/ pvkc
external f6,fpdfmf
if (pvkc.le.0.d0) go to 70
if (itcode-2) 10,20,30
10 v1=max(vc,vkmin)
whig=datan(higloo/de)-wphi0
if (whig.le.0.d0) go to 70
f7=pvkc-ggaus4(f6,v1,vkmax)
return
20 dth=datan(higloo/de)
if (indto.eq.1) go to 25
z=(log(dth)-zlgm)/zlgs
call mdnor(z,pz)
f7=pvkc*pz/zlgc
return
25 f7=pvkc*ggaus4(fpdfmf,0.d0,dth)
return
30 dth=datan(higloo/de)
z=0.d0
w=0.d0
do 50 i=1,5
if (dth.lt.gaphi(i)) go to 40
z=z+(gaphi(i)-w)*gaphia(i)
w=gaphi(i)
go to 50
40 z=z+(dth-w)*gaphia(i)
go to 60
50 continue
60 f7=pvkc*z
return
70 f7=0.d0
return
end
double precision function f8(th)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
alp=th-zmu
de=getde(alp)
de=dsqrt(de)
de2=de+de
f8=fpdfth(th)*f3()
return
end

```

```

double precision function f9(th)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
alp=th-zmu
de=getde(alp)
de=dsqrt(de)
de2=de+de
f9=fpdfth(th)*f7()
return
end
double precision function f10(thi)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
alp=thi-thmui
de=getde(alp)
de=dsqrt(de)
de2=de+de
f10=fpdfthsi(thi)*f4()
return
end
double precision function f11(the)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
alp=the-thmue
de=getde(alp)
de=dsqrt(de)
de2=de+de
f11=fpdfthen(the)*f4()
return
end
double precision function f12(a)
implicit double precision (a-h,o-z)
common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1p5max(3,2),cp5(3,2)
common /zlog/ zlmm,zlms,zlmc,zlgm,zlgs,zlgc,zlogm(3,2),
1zlogsd(3,2),zlogc(2)
external fp5
vl=fvlow(a)
if (vl.ge.p5mx) go to 10
y=(log(a)-zlmm)/zlms
f12=(dexp(-0.5d0*y*y)/(a*zlmc))*ggaus5(fp5,vl,p5mx)
return
10 f12=0.d0
return
end
double precision function f13(phi)
implicit double precision (a-h,o-z)
common /pars/ xmin,xmax,pside1,pendd,vc,vc2
common /gas/ dcf,cphig,cvkg
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
external f14
dcf=dcos(phi)
v1=vc/DCF
v2=gav(5)
if (v1.ge.v2) go to 10
f13=fpdfphi(phi)*ggaus5(f14,v1,v2)
return
10 f13=0.d0

```

```

return
end
double precision function fl4(v)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
common /gas/ dcf,cphig,cvkg
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /gask/ gaskmu(6),gaskvk(5),gaskphi(5),gasksig
vf=vc/(v*dcf)
s=de/(1.d0-vf*vf)
fvk=fpdfvkg(v)
z=(log(s)-gaskmu(igas)-cphig-cvkg)/gasksig
call mdnor(z,zp)
fl4=fvk*(1.d0-zp)
return
end
double precision function getde(alp)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
al=dsin(alp)
a1=a1*a1
a2=delam2+1.d0-2.d0*a1-2.d0*dcos(alp)*dsqrt(delam2-a1)
getde=d2*a2
return
end
double precision function fvlow(a)
implicit double precision (a-h,o-z)
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
fvlow=dsqrt(vc2+a*de2)
return
end
double precision function fcdfpv(w,v)
implicit double precision (a-h,o-z)
common /wphiv/ wphiv(3),wphivsh,wphi0
common /wpwp/ nwpv
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
if (w.le.wphi0) go to 10
if (w.ge.hpi) go to 20
av=dexp(weibp(v,nwpv,wphiv))
d1=dexp(-(wphi0/av)**wphivsh)
fcdfpv=(d1-dexp(-(w/av)**wphivsh))/(d1-dexp(-(hpi/av)**wphivsh))
return
10 fcdfpv=0.d0
return
20 fcdfpv=1.d0
return
end
double precision function weibp(v,nw,wp)
implicit double precision (a-h,o-z)
dimension wp(nw)
go to (10,20,30),nw
10 weibp=wp(1)
return
20 weibp=wp(1)+v*wp(2)
return
30 weibp=wp(1)+v*(wp(2)+v*wp(3))

```

```

return
end
subroutine prept(inds,th1,th2,th3,th4,ind2)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igac1,igac,igas
common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
common /xx/ x0,xdabs,xd2,indx
common /yy/ y0,alpha,beta,iout
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
common /phase/ indto
common /don/ aldon,bedon
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
lhigloo,rigloo,rac
ind2=0
if ((indto.eq.1).and.(itcode.eq.1)) go to 60
if (inds.ne.indx) go to 20
if (inds.eq.iout) go to 10
th1=alpha-beta
th2=alpha+beta
thmu1=alpha
return
10 th1=-alpha-beta
th2=-alpha+beta
thmu1=-alpha
return
20 if (inds.ne.iout) go to 40
if (alpha.lt.beta) go to 30
thmu1=alpha-pi
th1=thmu1-beta
th2=thmu1+beta
return
30 thmu1=alpha-pi
thmu2=alpha+pi
th1=-pi
th2=thmu1+beta
th3=thmu2-beta
th4=pi
ind2=1
return
40 if (alpha.lt.beta) go to 50
thmu1=-alpha+pi
th1=thmu1-beta
th2=thmu1+beta
return
50 thmu1=-alpha-pi
thmu2=-alpha+pi
th1=-pi
th2=thmu1+beta
th3=thmu2-beta
th4=pi
ind2=1
return
60 qx=qrw1-x0
if (qx.gt.0.d0) go to 70
thet0=hpi
go to 80
70 thet0=datan(abs(y0)/qx)
80 thet0=max(aldon+bedon*log(thet0),0.d0)

```



```

        if (inds.ne.indx) go to 100
        if (inds.eq.iout) go to 90
        ta=alpha-beta
        tb=alpha+beta
        go to 120
90      ta=-alpha-beta
        tb=-alpha+beta
        go to 120
100     if (inds.ne.iout) go to 110
        ta=alpha-beta-pi
        tb=alpha+beta-pi
        go to 120
110     ta=-alpha-beta+pi
        tb=-alpha+beta+pi
120     ta=ta-thet0
        tb=tb-thet0
        if (tb.gt.pin) go to 130
        ta=ta+pi2
        tb=tb+pi2
130     if (ta.lt.pin) go to 140
        if (tb.gt.pi) go to 150
        th1=ta
        th2=tb
        thmu1=0.5d0*(th1+th2)
        return
140     th1=-pi
        th2=tb
        th3=ta+pi2
        th4=pi
        ind2=1
        thmu1=0.5d0*(ta+tb)
        thmu2=thmu1+pi2
        return
150     th1=-pi
        th2=tb-pi2
        th3=ta
        th4=pi
        ind2=1
        thmu2=0.5d0*(ta+tb)
        thmu1=thmu2-pi2
        return
        end
        subroutine prepts(th1,th2)
        implicit double precision (a-h,o-z)
        common /xx/ x0,xdabs,xd2,indx
        common /yy/ y0,alpha,beta,iout
        common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
        common /tt/ thmul,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
        if (indx.eq.0) go to 20
        if (iout.eq.1) go to 10
        a=alpha-beta
        if (a.le.0.d0) go to 5
        th1=a
        th2=alpha+beta
        go to 40
5       th1=0.d0
        th2=alpha+beta
        go to 40

```

```

10  a=-alpha+beta
    if (a.le.0.d0) go to 15
    th1=0.d0
    th2=a
    go to 40
15  th1=0.d0
    th2=0.d0
    go to 40
20  if (iout.eq.1) go to 30
    a=-alpha+beta+pi
    if (a.ge.pi) go to 25
    th1=-alpha-beta+pi
    th2=a
    go to 40
25  th1=-alpha-beta+pi
    th2=pi
    go to 40
30  a=alpha-beta+pi
    if (a.ge.pi) go to 35
    th1=a
    th2=pi
    go to 40
35  th1=pi
    th2=pi
40  thmui=0.5d0*(th1+th2)
    return
    end
    subroutine prepte(th1,th2)
    implicit double precision (a-h,o-z)
    common /xx/ x0,xdabs,xd2,indx
    common /yy/ y0,alpha,beta,iout
    common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
    common /tt/ thmu1,thmu2,d2,delam2,de,de2,thmui,thmue,zmu
    if (indx.eq.0) go to 20
    if (iout.eq.1) go to 10
    a=alpha+beta
    if (a.ge.hpi) go to 5
    th1=alpha-beta
    th2=a
    go to 40
5    th1=alpha-beta
    th2=hpi
    go to 40
10   a=-alpha-beta
    if (a.le.hpin) go to 15
    th1=a
    th2=-alpha+beta
    go to 40
15   th1=-hpi
    th2=-alpha+beta
    go to 40
20   if (iout.eq.1) go to 30
    a=-alpha-beta+pi
    if (a.ge.hpi) go to 25
    th1=a
    th2=hpi
    go to 40
25   th1=hpi

```

```

th2=hpi
go to 40
30 a=alpha+beta-pi
   if (a.le.hpin) go to 35
   th1=hpin
   th2=a
   go to 40
35 th1=hpin
   th2=hpin
40 thmue=0.5d0*(th1+th2)
   return
end
double precision function gside(x)
implicit double precision (a-h,o-z)
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
external f10
call prepx(x)
call prepy(halfrw)
call prepts(thsi1,thsi2)
a1=ggaus2(f10,thsi1,thsi2)
gside=fpdfxs(x)*a1
return
end
double precision function fpdfxs(x)
implicit double precision (a-h,o-z)
common /igloo/ xigloo,yigloo,xtmb,xtpb,ytmb,qrw1,halfrw,wpr,wpr2,
1higloo,rigloo,rac
common /xside/ clside,c2side,c3side
xq=x/qrw1
fpdfxs=clside*(xq**c2side)*((1.d0-xq)**c3side)
return
end
double precision function gend(y)
implicit double precision (a-h,o-z)
external f11
call prepy(y)
call prepte(then1,then2)
a1=ggaus2(f11,then1,then2)
gend=fpdfye(y)*a1
return
end
double precision function fpdfye(y)
implicit double precision (a-h,o-z)
common /yend/ clend,c2end,c3end
fpdfye=clend*(c2end-y*y)**c3end
return
end
double precision function fpdfth(t)
implicit double precision (a-h,o-z)
common /codes/ itcode,imilc,igact,igacl,igac,igas
common /milth/ zmhm,zmhsd,zmhc,zmilhm(2,2),zmilhs(2,2),zmilhc(2,2)
common /betasg/ clland, c2land,c3land,clto,c2to,c3to
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
common /phase/ indto
if (itcode.eq.2) go to 20
x=rpi2*(t+pi)
if (indto.eq.itcode) go to 10

```

```

fpdfth=c1land*(x**c2land)*((1.d0-x)**c3land)
return
10 fpdfth=c1to*(x**c2to)*((1.d0-x)**c3to)
return
20 x=(t-zmhm)/zmhsd
fpdfth=dexp(-0.5d0*x*x)/zmhc
return
end
double precision function fpdfthsi(t)
implicit double precision (a-h,o-z)
common /betase/ t1side, t2side,t3side,t1end,t2end,t3end
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
x=rpi*t
fpdfthsi=t1side*(x**t2side)*((1.d0-x)**t3side)
return
end
double precision function fpdfthen(t)
implicit double precision (a-h,o-z)
common /betase/ t1side, t2side,t3side,t1end,t2end,t3end
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
x=rpi*(t+hpi)
fpdfthen=t1end*(x**t2end)*((1.d0-x)**t3end)
return
end
double precision function fpdfmf(phi)
implicit double precision (a-h,o-z)
common /const/ pi,hpi,pi2,pin,hpin,rpi,rpi2,spi
common /milphi/ c1pm,c2pm,c3pm,c1phim(2),c2phim(2),c3phim(2)
x=phi/hpi
fpdfmf=c1pm*(x**c2pm)*((1.d0-x)**c3pm)
return
end
double precision function fpdfml(y)
implicit double precision (a-h,o-z)
common /mill/ xmintm,ymintm,xminlm,yminlm,xmaxtm,ymaxtm,xmaxlm,
1 ymaxlm,flmil(16,16,2,2),cmil,ymilm(2),ymaxm(2)
common /flip/ iyneg(10,5),iflip
common /xx/ x0,xdabs,xd2,indx
common /codes/ itcode,imilc,igact,igacl,igac,igas
common /phase/ indto
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,ycx,yxc1,yxk,yxkca
common /pars/ xmin,xmax,psidel,pendd,vc,vc2
ii=(x0-xmin)/cmil+1.d0
if (iflip.eq.1) go to 10
jj=(y-ymin)/cmil+1.d0
5 fpdfml=flmil(ii,jj,imilc,indto)
return
10 jj=(ymax-y)/cmil+1.d0
go to 5
end
double precision function fpdfgl(y)
implicit double precision (a-h,o-z)
common /genl/ xming,yming,xmaxg,ymaxg,flgen(20,20,5)
common /flip/ iyneg(10,5),iflip
common /xx/ x0,xdabs,xd2,indx
common /codes/ itcode,imilc,igact,igacl,igac,igas
common /yx/ yxz1,yxz1p,yxz1m,ymin,ymax,yxa,ycx,yxc1,yxk,yxkca
common /pars/ xmin,xmax,psidel,pendd,vc,vc2

```

```

ii=(x0-xmin)/5280.d0+1.d0
if (iflip.eq.1) go to 10
jj=(y-ymin)/5280.d0+1.d0
5  fpdfgl=flgen(ii,jj,igac)
   return
10  jj=(ymax-y)/5280.d0+1.d0
   go to 5
end
double precision function fpdfphi(phi)
implicit double precision (a-h,o-z)
common /gas/ dcf,cphig,cvkg
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
common /gask/ gaskmu(6),gaskvk(5),gaskphi(5),gasksig
do 10 i=1,5
if (phi.le.gaphi(i)) go to 20
10  continue
20  cphig=gaskphi(i)
   fpdfphi=gaphia(i)
   return
end
double precision function fpdfvkg(v)
implicit double precision (a-h,o-z)
common /gas/ dcf,cphig,cvkg
common /gapv/ gaphi(5),gaphia(5),gav(5),gavb(5)
common /gask/ gaskmu(6),gaskvk(5),gaskphi(5),gasksig
do 10 i=1,5
if (v.le.gav(i)) go to 20
10  continue
20  cvkg=gaskvk(i)
   fpdfvkg=gavb(i)
   return
end
double precision function fp5(x)
implicit double precision (a-h,o-z)
common /pear5/ pear5a,pear5b,p5mx,comp5,p5a(3,2),p5b(3,2),
1p5max(3,2),cp5(3,2)
y=x/pear5b
fp5=1.d0/(comp5*(y**pear5a)*dexp(1.d0/y))
return
end
double precision function fip5(x)
implicit double precision (a-h,o-z)
common /flyin5/ fip5a,fip5b,fip5mx,fip5c
y=x/fip5b
fip5=1.d0/(fip5c*(y**fip5a)*dexp(1.d0/y))
return
end
double precision function fp6(x)
implicit double precision (a-h,o-z)
common /pear6/ pear6a,pear6b,pear6c,p6mx,comp6,p6a(2),p6b(2),
1p6c(2),p6max(2),cp6(2)
y=x/pear6c
fp6=(y**pear6a)/(comp6*(1.d0+y)**pear6b)
return
end
double precision function bgama(a,b)
implicit double precision (a-h,o-z)
bgama=dexp(algama(a+b)-algama(a)-algama(b))

```

```

return
end
double precision function acus(x)
implicit double precision (a-h,o-z)
if (x.lt.1.d0) go to 10
acus=0.d0
return
10  acus=dacos(x)
return
end
subroutine mdnor(y,p)
implicit double precision (a-h,o-z)
data sqrl2/.70710678118655d0/
p=0.5d0*erfcc(-y*sqrl2)
return
end
double precision function erfcc(x)
implicit double precision (a-h,o-z)
z=abs(x)
t=1.d0/(1.d0+0.5d0*z)
erfcc=t*dexp(-z*z-1.26551223d0+t*(1.00002368d0+t*(.37409196d0
1+t*(.09678418d0+t*(-.18628806d0+t*(.27886807d0+t*(-1.13520398d0
2+t*(1.48851587d0+t*(-.82215223d0+.17087277d0*t)))))))))
if (x.lt.0.d0) erfcc=2.d0-erfcc
return
end
double precision function algama(xx)
implicit double precision (a-h,o-z)
dimension cof(6)
save cof,stp
data cof,stp/76.18009172947146d0,-86.50532032941677d0,
124.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
2-.5395239384953d-5,2.5066282746310005d0/
x=xx
y=x
tmp=x+5.5d0
tmp=(x+0.5d0)*log(tmp)-tmp
ser=1.000000000190015d0
do 10 j=1,6
y=y+1.d0
ser=ser+cof(j)/y
10  continue
algama=tmp+log(stp*ser/x)
return
end
double precision function ggaus1(func,a,b)
implicit double precision (a-h,o-z)
external func
dimension w(5),x(5)
save w,x
data w/.2955242247d0,.2692667193d0,.2190863625d0,.1494513492d0,
1.0666713443d0/
data x/.1488743390d0,.4333953941d0,.6794095683d0,.8650633667d0,
1.9739065285d0/
xm=0.5d0*(b+a)
xr=0.5d0*(b-a)
ss=0.d0
do 10 j=1,5

```

```

dx=xr*x(j)
ss=ss+w(j)*(func(xm+dx)+func(xm-dx))
10 continue
ggaus1=xr*ss
return
end
double precision function ggaus2(func,a,b)
implicit double precision (a-h,o-z)
external func
dimension w(5),x(5)
save w,x
data w/.2955242247d0,.2692667193d0,.2190863625d0,.1494513492d0,
1.0666713443d0/
data x/.1488743390d0,.4333953941d0,.6794095683d0,.8650633667d0,
1.9739065285d0/
xm=0.5d0*(b+a)
xr=0.5d0*(b-a)
ss=0.d0
do 10 j=1,5
dx=xr*x(j)
ss=ss+w(j)*(func(xm+dx)+func(xm-dx))
10 continue
ggaus2=xr*ss
return
end
double precision function ggaus3(func,a,b)
implicit double precision (a-h,o-z)
external func
dimension w(5),x(5)
save w,x
data w/.2955242247d0,.2692667193d0,.2190863625d0,.1494513492d0,
1.0666713443d0/
data x/.1488743390d0,.4333953941d0,.6794095683d0,.8650633667d0,
1.9739065285d0/
xm=0.5d0*(b+a)
xr=0.5d0*(b-a)
ss=0.d0
do 10 j=1,5
dx=xr*x(j)
ss=ss+w(j)*(func(xm+dx)+func(xm-dx))
10 continue
ggaus3=xr*ss
return
end
double precision function ggaus4(func,a,b)
implicit double precision (a-h,o-z)
external func
dimension w(5),x(5)
save w,x
data w/.2955242247d0,.2692667193d0,.2190863625d0,.1494513492d0,
1.0666713443d0/
data x/.1488743390d0,.4333953941d0,.6794095683d0,.8650633667d0,
1.9739065285d0/
xm=0.5d0*(b+a)
xr=0.5d0*(b-a)
ss=0.d0
do 10 j=1,5
dx=xr*x(j)

```

```

10      ss=ss+w(j)*(func(xm+dx)+func(xm-dx))
        continue
        _ggaus4=xr*ss
        return
        end
        double precision function ggaus5(func,a,b)
        implicit double precision (a-h,o-z)
        external func
        dimension w(5),x(5)
        save w,x
        data w/.2955242247d0,.2692667193d0,.2190863625d0,.1494513492d0,
1.0666713443d0/
        data x/.1488743390d0,.4333953941d0,.6794095683d0,.8650633667d0,
1.9739065285d0/
        xm=0.5d0*(b+a)
        xr=0.5d0*(b-a)
        ss=0.d0
        do 10 j=1,5
        dx=xr*x(j)
10      ss=ss+w(j)*(func(xm+dx)+func(xm-dx))
        continue
        ggaus5=xr*ss
        return
        end

```

4.2. The input file, "crin".

Below is a line by line description of the entries in the user-supplied input file for the computation of $P_W(E)$. Refer to subroutine initial. An operation is defined as a take-off or landing.

for each type of aircraft, i

(1) itype(i) an indicator of whether type i aircraft use this airport

itype = 1 (yes); itype = 0 (no)

The types, i, are numbered as follows:

i = 1 commercial air carrier

i = 2 commercial air taxi

i = 3 military large aircraft

i = 4 military small aircraft

i = 5 military helicopter

- i = 6 general aviation single-engine reciprocating, fixed wing
- i = 7 general aviation multiple-engine reciprocating, fixed wing
- i = 8 general aviation turboprop, fixed wing
- i = 9 general aviation turbojet, fixed wing
- i = 10 general aviation reciprocating engine, rotary wing
- i = 11 general aviation turbine engine, rotary wing

[Note: Hereafter ntype is the number of types that use the airport. This is the number of 1's in the itype array. The type index l then refers to the relative type number among the types represented at the airport. Thus if the itype array is (11010011000), then ntype = 5, and l=1 is commercial air carrier, l = 2 is commercial air taxi, l = 3 is military small aircraft, l = 4 is general aviation fixed multiple-engine, l = 5 is general aviation fixed turboprop.]

- (2) niglloo the number of structures to be investigated (*max: 10*)
- nrunway the number of runways at the airport (*max: 5*)

[Note: The code can be used with more than one airport; in this case "airport" means the collection of airports that pose a non-negligible risk to the structure, and nrunway is the total number of runways in this collection of airports.]

for each runway, j:

- (3) qrwlj(j) the runway length, in feet
- runwid(j) the runway width, in feet
- (4) pruse(j) proportion of operations which use j [$\sum_j \text{pruse}(j) = 1$]
- prto(j) proportion of operations which use j that are take-offs

prdir1(j) proportion of operations which use j that have intended
direction 1 (R to L)

for each aircraft type index, l , given j :

(5) $\text{prmo}(j,l)$ proportion of operations which use runway j and involve type index l [$\sum_l \text{prmo}(j,l) = 1$]

for each structure, i:

(6) higlooi(i) height of structure, in feet
 diam diameter of structure, in feet

for each runway, j , given i :

(7) xigloo x-coordinate of center of structure

 yigloo y-coordinate of center of structure

[Note: The y-coordinate should be positive unless there is military use of the runway and the structure is located on the non-pattern side of the runway.]

Example: qrwlj=10000, runwid=150, xigloo=-12000, yigloo=100

(8) ncrit the number of critical velocities desired (*max*: 5)

for each critical velocity, i:

- (9) $v_{crit}(i)$ the value of the critical velocity, in feet per second
- (10) $noper$ the total number of operations at the airport in the time window

A sample version of "crin" is given below.

```
1 0 0 1 0 1 0 0 0 0 0
1 2
10000.d0 150.d0
.4d0 .5d0 .8d0
.55d0
.25d0
.20d0
10000.d0 150.d0
.6d0 .5d0 .8d0
.25d0
.75d0
0.d0
18.d0 150.d0
-10400.d0 650.d0
-10800.d0 -1375.d0
2
0.d0 150.d0
61908
```

4.3. The input file, "cring".

Below is the input file "cring" which contains probability distribution parameters used in the computation of $P_W(E)$. It is not intended to be modified by the user.

```
160.d0
70.d0
165.d0
60.d0
50.d0
33.d0
44.d0
60.d0
45.d0
```

40.d0
 40.d0
 0.57d0
 1.d-7
 -80000.d0 9.3663d0 0.4913d0
 2
 5.9531d0 -9.955d-5
 0.5381d0
 0.d0 115000.d0 9.6607d0 0.4947d0
 2
 5.2936d0 5.6063d-5
 0.82406d0
 1.400d0 1.887d0 2.301d0 1.552d0
 2.317d0
 100.d0 500.d0 161.668d0 1.9509d0
 10.d0 350.d0 137.8d0 1.53d0
 1.d0 150.d0 3.190d0 0.5635d0
 1.d0 40.d0 2.2897d0 0.6134d0
 3
 1.8580d0 -2.2191d-2 3.6529d-5
 1.08931d0
 0.5d0
 0.97126d0 0.16906d0
 132.d0 132.d0
 61.18d0 56.76d0
 1.158d0 19.26d0
 104.0d0 96.91d0
 1.90d-07 3.40d-07 2.20d-07 3.80d-07 5.40d-06 3.00d-05
 5.70d-05 3.80d-05 8.60d-06 6.70d-07 1.80d-08 1.60d-10
 5.20d-13
 1.00d-05 1.90d-05 1.20d-05 8.00d-06 8.60d-05 4.80d-04
 9.20d-04 6.10d-04 1.40d-04 1.10d-05 2.90d-07 2.60d-09
 8.30d-12
 2.30d-04 4.20d-04 2.70d-04 9.70d-05 5.50d-04 3.02d-03
 5.78d-03 3.81d-03 8.60d-04 6.70d-05 1.80d-06 1.70d-08
 5.20d-11
 1.98d-03 3.79d-03 2.62d-03 8.10d-04 1.45d-03 7.46d-03
 1.43d-02 9.41d-03 2.13d-03 1.70d-04 4.40d-06 4.10d-08
 1.30d-10
 7.22d-03 1.50d-02 1.23d-02 4.89d-03 2.55d-03 7.42d-03
 1.39d-02 9.12d-03 2.07d-03 1.60d-04 4.30d-06 4.00d-08
 1.20d-10
 1.24d-02 3.19d-02 3.54d-02 2.00d-02 7.38d-03 4.00d-03
 5.35d-03 3.47d-03 7.80d-04 6.10d-05 1.60d-06 1.50d-08
 4.70d-11
 1.29d-02 4.44d-02 6.54d-02 4.72d-02 1.88d-02 3.97d-03
 1.05d-03 5.20d-04 1.20d-04 9.10d-06 2.40d-07 2.20d-09
 7.10d-12
 9.65d-03 4.16d-02 7.54d-02 6.68d-02 3.05d-02 6.28d-03
 5.40d-04 4.50d-05 7.00d-06 5.30d-07 1.40d-08 1.30d-10
 4.10d-13
 5.22d-03 2.57d-02 5.59d-02 6.19d-02 3.37d-02 7.66d-03
 6.50d-04 2.00d-05 3.70d-07 1.30d-08 3.30d-10 3.00d-12
 9.50d-15
 1.78d-03 9.56d-03 2.44d-02 3.29d-02 2.06d-02 5.04d-03
 4.50d-04 1.40d-05 1.50d-07 6.80d-10 3.70d-12 2.70d-14
 8.50d-17

3.10d-04	1.79d-03	5.28d-03	8.27d-03	5.65d-03	1.45d-03	
1.30d-04	4.20d-06	4.60d-08	1.80d-10	2.40d-13	2.00d-16	
3.20d-19						
7.00d-05	4.00d-04	1.00d-03	3.00d-03	7.00d-03	9.00d-03	
9.00d-03	6.00d-03	3.00d-03	1.00d-03	3.00d-04	7.00d-05	
5.00d-05	6.00d-05	4.00d-05	3.00d-05			
2.00d-04	1.00d-03	4.00d-03	9.00d-03	1.60d-02	2.00d-02	
1.80d-02	1.20d-02	5.00d-03	2.00d-03	5.00d-04	3.00d-04	
4.00d-04	5.00d-04	4.00d-04	2.00d-04			
4.00d-04	2.00d-03	6.00d-03	1.50d-02	2.70d-02	3.30d-02	
2.90d-02	1.80d-02	8.00d-03	3.00d-03	1.00d-03	1.00d-03	
2.00d-03	2.00d-03	2.00d-03	9.00d-04			
5.00d-04	2.00d-03	7.00d-03	1.70d-02	2.90d-02	3.60d-02	
3.20d-02	2.00d-02	9.00d-03	3.00d-03	2.00d-03	2.00d-03	
4.00d-03	4.00d-03	3.00d-03	2.00d-03			
7.00d-04	2.00d-03	7.00d-03	1.50d-02	2.40d-02	2.80d-02	
2.50d-02	1.50d-02	7.00d-03	3.00d-03	2.00d-03	3.00d-03	
4.00d-03	5.00d-03	4.00d-03	2.00d-03			
1.00d-03	3.00d-03	8.00d-03	1.40d-02	2.00d-02	2.30d-02	
1.90d-02	1.10d-02	5.00d-03	2.00d-03	2.00d-03	3.00d-03	
4.00d-03	4.00d-03	3.00d-03	2.00d-03			
1.00d-03	3.00d-03	7.00d-03	1.30d-02	1.80d-02	2.00d-02	
1.60d-02	1.00d-02	5.00d-03	2.00d-03	2.00d-03	3.00d-03	
5.00d-03	4.00d-03	3.00d-03	1.00d-03			
7.00d-04	2.00d-03	4.00d-03	8.00d-03	1.20d-02	1.40d-02	
1.20d-02	8.00d-03	4.00d-03	2.00d-03	2.00d-03	4.00d-03	
5.00d-03	4.00d-03	3.00d-03	1.00d-03			
2.00d-04	7.00d-04	2.00d-03	3.00d-03	5.00d-03	6.00d-03	
6.00d-03	4.00d-03	2.00d-03	1.00d-03	1.00d-03	2.00d-03	
3.00d-03	3.00d-03	2.00d-03	8.00d-04			
4.00d-05	1.00d-04	3.00d-04	7.00d-04	1.00d-03	2.00d-03	
2.00d-03	1.00d-03	6.00d-04	4.00d-04	4.00d-04	7.00d-04	
9.00d-04	8.00d-04	5.00d-04	2.00d-04			
4.00d-06	1.00d-05	4.00d-05	9.00d-05	2.00d-04	3.00d-04	
3.00d-04	2.00d-04	1.00d-04	6.00d-05	7.00d-05	1.00d-04	
1.00d-04	1.00d-04	9.00d-05	4.00d-05			
2.00d-07	8.00d-07	3.00d-06	7.00d-06	1.00d-05	2.00d-05	
2.00d-05	2.00d-05	9.00d-06	5.00d-06	6.00d-06	9.00d-06	
1.00d-05	1.00d-05	7.00d-06	3.00d-06			
6.00d-09	3.00d-08	1.00d-07	3.00d-07	6.00d-07	1.00d-06	
1.00d-06	8.00d-07	4.00d-07	2.00d-07	2.00d-07	4.00d-07	
5.00d-07	5.00d-07	3.00d-07	1.00d-07			
1.00d-10	6.00d-10	2.00d-09	7.00d-09	2.00d-08	2.00d-08	
3.00d-08	2.00d-08	1.00d-08	6.00d-09	6.00d-09	9.00d-09	
1.00d-08	1.00d-08	7.00d-09	3.00d-09			
1.00d-12	7.00d-12	3.00d-11	9.00d-11	2.00d-10	3.00d-10	
3.00d-10	2.00d-10	1.00d-10	7.00d-11	7.00d-11	1.00d-10	
1.00d-10	1.00d-10	8.00d-11	4.00d-11			
7.00d-15	4.00d-14	2.00d-13	6.00d-13	1.00d-12	2.00d-12	
2.00d-12	2.00d-12	9.00d-13	4.00d-13	4.00d-13	6.00d-13	
8.00d-13	7.00d-13	5.00d-13	2.00d-13			
3.00d-04	1.10d-03	2.71d-03	3.96d-03	3.75d-03	2.95d-03	
1.88d-03	8.00d-04	2.30d-04	5.00d-05	7.70d-06	7.30d-07	
4.20d-08						
8.10d-04	2.01d-03	3.58d-03	4.24d-03	3.26d-03	2.23d-03	
1.76d-03	1.19d-03	5.70d-04	1.80d-04	3.30d-05	3.60d-06	
3.00d-07						

1.97d-03	4.11d-03	5.77d-03	5.51d-03	3.61d-03	2.62d-03	
2.89d-03	2.60d-03	1.48d-03	5.00d-04	1.00d-04	1.50d-05	
2.00d-06						
3.50d-03	7.08d-03	9.47d-03	8.44d-03	5.18d-03	3.83d-03	
4.70d-03	4.60d-03	2.77d-03	1.00d-03	2.50d-04	5.90d-05	
1.20d-05						
4.55d-03	9.75d-03	1.39d-02	1.28d-02	7.82d-03	5.10d-03	
5.78d-03	5.81d-03	3.66d-03	1.51d-03	5.60d-04	2.10d-04	
5.40d-05						
4.91d-03	1.19d-02	1.83d-02	1.77d-02	1.10d-02	6.23d-03	
5.62d-03	5.48d-03	3.71d-03	2.00d-03	1.19d-03	6.00d-04	
1.80d-04						
5.45d-03	1.49d-02	2.40d-02	2.32d-02	1.46d-02	7.70d-03	
5.10d-03	4.39d-03	3.36d-03	2.56d-03	2.11d-03	1.27d-03	
4.60d-04						
6.69d-03	1.96d-02	3.15d-02	2.93d-02	1.82d-02	9.33d-03	
4.93d-03	3.57d-03	3.13d-03	2.98d-03	2.89d-03	2.04d-03	
8.70d-04						
7.76d-03	2.29d-02	3.63d-02	3.25d-02	1.92d-02	9.50d-03	
4.52d-03	3.00d-03	2.82d-03	2.80d-03	2.97d-03	2.44d-03	
1.18d-03						
7.35d-03	2.13d-02	3.29d-02	2.82d-02	1.56d-02	7.23d-03	
3.25d-03	2.14d-03	2.05d-03	1.98d-03	2.28d-03	2.12d-03	
1.12d-03						
5.27d-03	1.48d-02	2.20d-02	1.80d-02	9.24d-03	3.90d-03	
1.66d-03	1.11d-03	1.07d-03	1.02d-03	1.27d-03	1.29d-03	
7.10d-04						
3.00d-07	1.00d-05	1.00d-04	2.00d-03	1.10d-02	2.20d-02	
1.30d-02	3.00d-03	8.00d-04	2.00d-04	1.00d-05	2.00d-07	
2.00d-08	4.00d-08	4.00d-08	1.00d-08			
9.00d-07	3.00d-05	4.00d-04	3.00d-03	2.10d-02	4.50d-02	
2.70d-02	6.00d-03	1.00d-03	3.00d-04	2.00d-05	1.00d-06	
3.00d-07	6.00d-07	6.00d-07	2.00d-07			
2.00d-06	6.00d-05	6.00d-04	5.00d-03	3.10d-02	6.50d-02	
3.90d-02	8.00d-03	2.00d-03	4.00d-04	6.00d-05	7.00d-06	
3.00d-06	6.00d-06	6.00d-06	2.00d-06			
2.00d-06	7.00d-05	7.00d-04	5.00d-03	3.20d-02	6.90d-02	
4.20d-02	9.00d-03	2.00d-03	7.00d-04	2.00d-04	4.00d-05	
2.00d-05	4.00d-05	4.00d-05	2.00d-05			
2.00d-06	6.00d-05	6.00d-04	4.00d-03	2.50d-02	5.50d-02	
3.40d-02	8.00d-03	2.00d-03	1.00d-03	4.00d-04	2.00d-04	
8.00d-05	2.00d-04	2.00d-04	9.00d-05			
8.00d-06	4.00d-05	4.00d-04	3.00d-03	1.60d-02	3.50d-02	
2.20d-02	6.00d-03	2.00d-03	2.00d-03	1.00d-03	5.00d-04	
2.00d-04	5.00d-04	6.00d-04	3.00d-04			
5.00d-05	5.00d-05	4.00d-04	2.00d-03	1.00d-02	2.00d-02	
1.30d-02	4.00d-03	2.00d-03	2.00d-03	2.00d-03	1.00d-03	
5.00d-04	1.00d-03	1.00d-03	8.00d-04			
2.00d-04	1.00d-04	4.00d-04	2.00d-03	7.00d-03	1.20d-02	
8.00d-03	3.00d-03	1.00d-03	2.00d-03	2.00d-03	2.00d-03	
6.00d-04	1.00d-03	2.00d-03	1.00d-03			
6.00d-04	4.00d-04	5.00d-04	2.00d-03	5.00d-03	8.00d-03	
6.00d-03	2.00d-03	8.00d-04	1.00d-03	2.00d-03	1.00d-03	
6.00d-04	1.00d-03	2.00d-03	1.00d-03			
1.00d-03	7.00d-04	6.00d-04	2.00d-03	5.00d-03	7.00d-03	
5.00d-03	2.00d-03	4.00d-04	8.00d-04	2.00d-03	1.00d-03	
4.00d-04	6.00d-04	1.00d-03	1.00d-03			

1.00d-03	8.00d-04	6.00d-04	2.00d-03	4.00d-03	6.00d-03	
5.00d-03	2.00d-03	3.00d-04	4.00d-04	1.00d-03	1.00d-03	
4.00d-04	2.00d-04	4.00d-04	5.00d-04			
1.00d-03	6.00d-04	5.00d-04	2.00d-03	3.00d-03	5.00d-03	
5.00d-03	2.00d-03	2.00d-04	3.00d-04	1.00d-03	1.00d-03	
4.00d-04	9.00d-05	1.00d-04	2.00d-04			
6.00d-04	3.00d-04	2.00d-04	9.00d-04	2.00d-03	5.00d-03	
5.00d-03	2.00d-03	2.00d-04	2.00d-04	9.00d-04	1.00d-03	
4.00d-04	4.00d-05	2.00d-05	4.00d-05			
2.00d-04	1.00d-04	8.00d-05	4.00d-04	2.00d-03	5.00d-03	
5.00d-03	2.00d-03	2.00d-04	1.00d-04	6.00d-04	7.00d-04	
2.00d-04	2.00d-05	4.00d-06	5.00d-06			
5.00d-05	3.00d-05	2.00d-05	3.00d-04	2.00d-03	5.00d-03	
4.00d-03	1.00d-03	1.00d-04	5.00d-05	2.00d-04	3.00d-04	
1.00d-04	8.00d-06	5.00d-07	5.00d-07			
7.00d-06	4.00d-06	8.00d-06	2.00d-04	1.00d-03	3.00d-03	
2.00d-03	7.00d-04	8.00d-05	2.00d-05	7.00d-05	9.00d-05	
3.00d-05	2.00d-06	6.00d-08	3.00d-08			
0.0d+0	0.0d+0	0.0d+0	0.0d+0	4.1d-5	1.5d-4	
4.3d-4	9.4d-4	1.6d-3	2.0d-3	2.0d-3	1.5d-3	
9.0d-4	4.3d-4	1.7d-4	7.3d-5	3.5d-5	1.8d-5	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	1.1d-5	5.4d-5	2.0d-4	
5.8d-4	1.3d-3	2.2d-3	2.8d-3	2.8d-3	2.2d-3	
1.3d-3	6.0d-4	2.5d-4	1.0d-4	5.1d-5	2.6d-5	
1.1d-5	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	1.4d-5	7.0d-5	2.7d-4	
7.7d-4	1.7d-3	2.9d-3	3.8d-3	3.8d-3	2.9d-3	
1.7d-3	8.2d-4	3.4d-4	1.5d-4	7.1d-5	3.6d-5	
1.5d-5	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	1.8d-5	8.9d-5	3.4d-4	
9.9d-4	2.2d-3	3.8d-3	5.0d-3	5.0d-3	3.8d-3	
2.3d-3	1.1d-3	4.5d-4	1.9d-4	9.7d-5	4.8d-5	
2.1d-5	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	2.3d-5	1.1d-4	4.2d-4	
1.2d-3	2.8d-3	4.8d-3	6.3d-3	6.3d-3	4.9d-3	
2.9d-3	1.4d-3	5.8d-4	2.5d-4	1.3d-4	6.4d-5	
2.7d-5	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	2.7d-5	1.3d-4	5.1d-4	
1.5d-3	3.4d-3	5.8d-3	7.7d-3	7.8d-3	6.0d-3	
3.6d-3	1.7d-3	7.2d-4	3.2d-4	1.6d-4	8.1d-5	
3.5d-5	1.2d-5					
0.0d+0	0.0d+0	0.0d+0	3.2d-5	1.6d-4	6.0d-4	
1.8d-3	4.0d-3	6.9d-3	9.1d-3	9.2d-3	7.1d-3	
4.3d-3	2.1d-3	8.7d-4	3.8d-4	2.0d-4	9.9d-5	
4.3d-5	1.5d-5					
0.0d+0	0.0d+0	3.7d-5	1.8d-4	6.8d-4	2.0d-3	
4.6d-3	7.9d-3	1.0d-2	1.1d-2	8.2d-3	4.9d-3	
2.4d-3	1.0d-3	4.5d-4	2.3d-4	1.2d-4	5.1d-5	
1.7d-5	1.5d-5					
0.0d+0	0.0d+0	0.0d+0	4.1d-5	2.0d-4	7.6d-4	
2.2d-3	5.1d-3	8.8d-3	1.2d-2	1.2d-2	9.1d-3	
5.5d-3	2.7d-3	1.1d-3	5.1d-4	2.6d-4	1.3d-4	
5.8d-5	2.0d-5					
0.0d+0	0.0d+0	0.0d+0	4.5d-5	2.2d-4	8.1d-4	
2.4d-3	5.4d-3	9.4d-3	1.3d-2	1.3d-2	9.8d-3	
5.9d-3	2.9d-3	1.2d-3	5.6d-4	2.9d-4	1.5d-4	
6.4d-5	2.2d-5					

0.0d+0	0.0d+0	0.0d+0	4.8d-5	2.3d-4	8.5d-4	
2.5d-3	5.6d-3	9.8d-3	1.3d-2	1.3d-2	1.0d-2	
6.2d-3	3.0d-3	1.3d-3	6.0d-4	3.1d-4	1.6d-4	
6.9d-5	2.4d-5					
0.0d+0	0.0d+0	0.0d+0	4.9d-5	2.3d-4	8.5d-4	
2.5d-3	5.7d-3	9.8d-3	1.3d-2	1.3d-2	1.0d-2	
6.3d-3	3.1d-3	1.3d-3	6.1d-4	3.2d-4	1.6d-4	
7.1d-5	2.5d-5					
0.0d+0	0.0d+0	0.0d+0	5.0d-5	2.3d-4	8.4d-4	
2.4d-3	5.5d-3	9.6d-3	1.3d-2	1.3d-2	1.0d-2	
6.1d-3	3.0d-3	1.3d-3	6.1d-4	3.2d-4	1.6d-4	
7.1d-5	2.5d-5					
0.0d+0	0.0d+0	0.0d+0	4.8d-5	2.2d-4	7.9d-4	
2.3d-3	5.2d-3	9.0d-3	1.2d-2	1.2d-2	9.5d-3	
5.8d-3	2.8d-3	1.2d-3	5.8d-4	3.1d-4	1.6d-4	
6.8d-5	2.4d-5					
0.0d+0	0.0d+0	0.0d+0	4.6d-5	2.0d-4	7.3d-4	
2.1d-3	4.7d-3	8.2d-3	1.1d-2	1.1d-2	8.7d-3	
5.3d-3	2.6d-3	1.1d-3	5.4d-4	2.8d-4	1.5d-4	
6.4d-5	2.2d-5					
0.0d+0	0.0d+0	0.0d+0	4.2d-5	1.8d-4	6.5d-4	
1.9d-3	4.2d-3	7.2d-3	9.6d-3	9.8d-3	7.6d-3	
4.7d-3	2.3d-3	1.0d-3	4.8d-4	2.6d-4	1.3d-4	
5.7d-5	2.0d-5					
0.0d+0	0.0d+0	0.0d+0	3.8d-5	1.6d-4	5.7d-4	
1.6d-3	3.6d-3	6.1d-3	8.2d-3	8.3d-3	6.5d-3	
4.0d-3	2.0d-3	8.9d-4	4.2d-4	2.2d-4	1.2d-4	
5.0d-5	1.8d-5					
0.0d+0	0.0d+0	0.0d+0	3.3d-5	1.4d-4	4.7d-4	
1.3d-3	2.9d-3	5.1d-3	6.7d-3	6.9d-3	5.4d-3	
3.3d-3	1.7d-3	7.4d-4	3.5d-4	1.9d-4	9.6d-5	
4.2d-5	1.5d-5					
0.0d+0	0.0d+0	0.0d+0	2.8d-5	1.1d-4	3.9d-4	
1.1d-3	2.4d-3	4.0d-3	5.4d-3	5.5d-3	4.3d-3	
2.7d-3	1.3d-3	6.0d-4	2.9d-4	1.5d-4	7.8d-5	
3.4d-5	1.2d-5					
0.0d+0	0.0d+0	0.0d+0	2.3d-5	9.2d-5	3.1d-4	
8.4d-4	1.8d-3	3.1d-3	4.1d-3	4.2d-3	3.3d-3	
2.1d-3	1.0d-3	4.7d-4	2.2d-4	1.2d-4	6.1d-5	
2.7d-5	1.0d-5					
0.0d+0	0.0d+0	0.0d+0	2.0d-5	5.7d-5	1.3d-4	
2.8d-4	5.3d-4	8.9d-4	1.2d-3	1.3d-3	1.0d-3	
6.2d-4	3.0d-4	1.3d-4	6.8d-5	4.9d-5	5.0d-5	
6.4d-5	9.0d-5					
0.0d+0	0.0d+0	1.1d-5	3.4d-5	8.9d-5	1.9d-4	
3.8d-4	7.2d-4	1.7d-3	1.7d-3	1.3d-3	7.1d-4	
3.1d-4	1.2d-4	4.8d-5	2.8d-5	1.2d-3	2.8d-5	
4.0d-5	6.7d-5					
0.0d+0	0.0d+0	1.9d-5	5.5d-5	1.3d-4	2.8d-4	
5.4d-4	1.0d-3	1.8d-3	2.5d-3	2.5d-3	1.8d-3	
9.1d-4	3.5d-4	1.2d-4	4.3d-5	2.0d-5	1.5d-5	
2.3d-5	4.4d-5					
0.0d+0	1.2d-5	3.2d-5	8.3d-5	1.9d-4	3.9d-4	
7.7d-4	1.5d-3	2.8d-3	3.8d-3	3.8d-3	2.6d-3	
1.3d-3	4.9d-4	1.6d-4	6.0d-5	2.4d-5	1.2d-5	
1.3d-5	2.6d-5					
0.0d+0	2.0d-5	5.2d-5	1.2d-4	2.6d-4	5.3d-4	
1.1d-3	2.4d-3	4.4d-3	6.1d-3	6.0d-3	4.1d-3	

	2.0d-3	7.6d-4	2.7d-4	1.1d-4	4.5d-5	1.6d-5
	1.4d-5	0.0d+0				
1.0d-5	3.1d-5	7.5d-5	1.6d-4	3.3d-4	6.9d-4	
	1.6d-3	3.6d-3	6.8d-3	9.6d-3	9.5d-3	6.5d-3
	3.2d-3	1.2d-3	4.6d-4	2.0d-4	8.5d-5	2.9d-5
	1.4d-5	0.0d+0				
1.4d-5	4.2d-5	9.8d-5	2.0d-4	3.9d-4	8.6d-4	
	2.2d-3	5.2d-3	1.0d-2	1.4d-2	1.4d-2	9.9d-3
	4.9d-3	1.9d-3	7.3d-4	3.3d-4	1.4d-4	4.9d-5
	1.3d-5	0.0d+0				
1.5d-5	4.8d-5	1.1d-4	2.2d-4	4.4d-4	1.0d-3	
	2.8d-3	7.1d-3	1.4d-2	2.0d-2	2.0d-2	1.4d-2
	6.8d-3	2.6d-3	1.0d-3	4.7d-4	2.1d-4	7.4d-5
	1.8d-5	0.0d+0				
1.5d-5	4.8d-5	1.2d-4	2.4d-4	4.7d-4	1.1d-3	
	3.3d-3	8.7d-3	1.7d-2	2.5d-2	2.5d-2	1.7d-2
	8.4d-3	3.3d-3	1.3d-3	5.9d-4	2.7d-4	9.7d-5
	2.4d-5	0.0d+0				
1.3d-5	4.2d-5	1.1d-4	2.3d-4	4.7d-4	1.2d-3	
	3.5d-3	9.4d-3	1.9d-2	2.7d-2	2.7d-2	1.9d-2
	9.3d-3	3.6d-3	1.4d-3	6.6d-4	3.1d-4	1.1d-4
	2.8d-5	0.0d+0				
0.0d+0	3.3d-5	8.9d-5	2.0d-4	4.3d-4	1.1d-3	
	3.4d-3	9.1d-3	1.8d-2	2.6d-2	2.6d-2	1.8d-2
	9.0d-3	3.5d-3	1.4d-3	6.5d-4	3.0d-4	1.1d-4
	2.8d-5	0.0d+0				
0.0d+0	2.3d-5	6.9d-5	1.7d-4	3.7d-4	9.7d-4	
	2.9d-3	7.7d-3	1.6d-2	2.2d-2	2.2d-2	1.6d-2
	7.7d-3	3.0d-3	1.2d-3	5.7d-4	2.6d-4	9.5d-5
	2.4d-5	0.0d+0				
0.0d+0	1.5d-5	5.0d-5	1.3d-4	3.0d-4	7.7d-4	
	2.2d-3	5.9d-3	1.2d-2	1.7d-2	1.7d-2	1.2d-2
	5.9d-3	2.4d-3	9.5d-4	4.5d-4	2.1d-4	7.3d-5
	1.8d-5	0.0d+0				
0.0d+0	1.0d-5	3.6d-5	9.9d-5	2.3d-4	5.7d-4	
	1.6d-3	4.0d-3	8.0d-3	1.1d-2	1.1d-2	8.0d-3
	4.1d-3	1.7d-3	7.1d-4	3.4d-4	1.5d-4	5.0d-5
	1.2d-5	0.0d+0				
0.0d+0	0.0d+0	2.5d-5	7.3d-5	1.7d-4	4.0d-4	
	1.0d-3	2.5d-3	4.9d-3	7.0d-3	7.0d-3	5.0d-3
	2.6d-3	1.1d-3	5.0d-4	2.3d-4	9.7d-5	3.1d-5
	0.0d+0	0.0d+0				
0.0d+0	0.0d+0	1.7d-5	5.1d-5	1.2d-4	2.7d-4	
	6.3d-4	1.5d-3	2.8d-3	4.0d-3	4.1d-3	2.9d-3
	1.6d-3	7.4d-4	3.4d-4	1.5d-4	6.0d-5	1.8d-5
	0.0d+0	0.0d+0				
0.0d+0	0.0d+0	1.1d-5	3.4d-5	8.1d-5	1.7d-4	
	3.7d-4	8.3d-4	1.6d-3	2.2d-3	2.3d-3	1.7d-3
	9.5d-4	4.6d-4	2.1d-4	9.2d-5	3.4d-5	1.0d-5
	0.0d+0	0.0d+0				
0.0d+0	0.0d+0	0.0d+0	2.4d-5	5.6d-5	1.1d-4	
	2.2d-4	4.6d-4	8.4d-4	1.2d-3	1.2d-3	9.2d-4
	5.4d-4	2.7d-4	1.2d-4	5.1d-5	0.0d+0	0.0d+0
	0.0d+0	0.0d+0				
0.0d+0	0.0d+0	0.0d+0	2.0d-5	4.7d-5	8.3d-5	
	1.4d-4	2.5d-4	4.4d-4	6.2d-4	6.4d-4	5.0d-4
	3.0d-4	1.5d-4	6.5d-5	2.6d-5	0.0d+0	0.0d+0
	0.0d+0	0.0d+0				

0.0d+0	0.0d+0	0.0d+0	2.2d-5	4.9d-5	7.8d-5	
1.0d-4	1.4d-4	2.3d-4	3.2d-4	3.4d-4	2.7d-4	
1.7d-4	7.9d-5	3.2d-5	1.2d-5	0.0d+0	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	1.2d-5	3.4d-5	8.5d-5	
2.0d-4	3.9d-4	6.5d-4	8.7d-4	9.1d-4	7.5d-4	
4.8d-4	2.4d-4	9.5d-5	2.9d-5	0.0d+0	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	0.0d+0	2.1d-5	5.8d-5	1.4d-4	
3.3d-4	6.5d-4	1.1d-3	1.4d-3	1.5d-3	1.2d-3	
7.8d-4	3.9d-4	1.6d-4	4.8d-5	1.2d-5	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	1.1d-5	3.4d-5	9.4d-5	2.3d-4	
5.2d-4	1.0d-3	1.7d-3	2.2d-3	2.3d-3	1.9d-3	
1.2d-3	6.1d-4	2.4d-4	7.5d-5	1.8d-5	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	1.7d-5	5.4d-5	1.5d-4	3.6d-4	
7.9d-4	1.5d-3	2.5d-3	3.3d-3	3.5d-3	2.8d-3	
1.8d-3	9.1d-4	3.6d-4	1.1d-4	2.7d-5	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	2.5d-5	8.1d-5	2.2d-4	5.2d-4	
1.1d-3	2.2d-3	3.6d-3	4.7d-3	4.9d-3	4.0d-3	
2.6d-3	1.3d-3	5.1d-4	1.6d-4	3.9d-5	0.0d+0	
0.0d+0	0.0d+0					
0.0d+0	0.0d+0	3.7d-5	1.2d-4	3.1d-4	7.3d-4	
1.6d-3	3.0d-3	4.8d-3	6.3d-3	6.6d-3	5.4d-3	
3.5d-3	1.7d-3	6.9d-4	2.2d-4	5.3d-5	1.0d-5	
0.0d+0	0.0d+0					
0.0d+0	1.3d-5	5.0d-5	1.6d-4	4.2d-4	9.8d-4	
2.1d-3	3.9d-3	6.2d-3	8.2d-3	8.5d-3	7.0d-3	
4.5d-3	2.2d-3	8.9d-4	2.8d-4	6.8d-5	1.3d-5	
0.0d+0	0.0d+0					
0.0d+0	1.7d-5	6.6d-5	2.1d-4	5.4d-4	1.2d-3	
2.6d-3	4.8d-3	7.7d-3	1.0d-2	1.0d-2	8.5d-3	
5.5d-3	2.7d-3	1.1d-3	3.4d-4	8.3d-5	1.6d-5	
0.0d+0	0.0d+0					
0.0d+0	2.1d-5	8.2d-5	2.6d-4	6.7d-4	1.5d-3	
3.1d-3	5.7d-3	9.0d-3	1.2d-2	1.2d-2	9.9d-3	
6.4d-3	3.2d-3	1.3d-3	3.9d-4	9.7d-5	1.9d-5	
0.0d+0	0.0d+0					
0.0d+0	2.5d-5	9.7d-5	3.0d-4	7.8d-4	1.7d-3	
3.5d-3	6.4d-3	1.0d-2	1.3d-2	1.4d-2	1.1d-2	
7.1d-3	3.6d-3	1.4d-3	4.4d-4	1.1d-4	2.1d-5	
0.0d+0	0.0d+0					
0.0d+0	2.8d-5	1.1d-4	3.4d-4	8.7d-4	1.9d-3	
3.8d-3	6.9d-3	1.1d-2	1.4d-2	1.4d-2	1.2d-2	
7.5d-3	3.8d-3	1.5d-3	4.6d-4	1.1d-4	2.2d-5	
0.0d+0	0.0d+0					
0.0d+0	3.0d-5	1.2d-4	3.7d-4	9.3d-4	2.0d-3	
4.0d-3	7.1d-3	1.1d-2	1.4d-2	1.4d-2	1.2d-2	
7.6d-3	3.8d-3	1.5d-3	4.7d-4	1.2d-4	2.2d-5	
0.0d+0	0.0d+0					
0.0d+0	3.1d-5	1.2d-4	3.7d-4	9.4d-4	2.0d-3	
3.9d-3	6.9d-3	1.1d-2	1.4d-2	1.4d-2	1.1d-2	
7.3d-3	3.6d-3	1.4d-3	4.5d-4	1.1d-4	2.1d-5	
0.0d+0	0.0d+0					
0.0d+0	3.0d-5	1.2d-4	3.6d-4	9.1d-4	1.9d-3	
3.7d-3	6.4d-3	9.8d-3	1.2d-2	1.3d-2	1.0d-2	

	6.7d-3	3.3d-3	1.3d-3	4.1d-4	1.0d-4	2.0d-5
	0.0d+0	0.0d+0				
0.0d+0	2.8d-5	1.1d-4	3.4d-4	8.4d-4	1.8d-3	
	3.3d-3	5.7d-3	8.6d-3	1.1d-2	1.1d-2	9.1d-3
	5.8d-3	2.9d-3	1.2d-3	3.6d-4	8.8d-5	1.7d-5
	0.0d+0	0.0d+0				
0.0d+0	2.5d-5	9.7d-5	3.0d-4	7.4d-4	1.5d-3	
	2.9d-3	4.8d-3	7.3d-3	9.2d-3	9.4d-3	7.6d-3
	4.8d-3	2.4d-3	9.6d-4	3.0d-4	7.4d-5	1.5d-5
	0.0d+0	0.0d+0				
0.0d+0	2.1d-5	8.2d-5	2.5d-4	6.2d-4	1.3d-3	
	2.3d-3	3.9d-3	5.8d-3	7.3d-3	7.4d-3	6.0d-3
	3.8d-3	1.9d-3	7.6d-4	2.4d-4	5.9d-5	1.2d-5
	0.0d+0	0.0d+0				
0.0d+0	1.7d-5	6.6d-5	2.0d-4	4.9d-4	5.7d-4	
	1.0d-3	1.8d-3	3.0d-3	4.5d-3	5.6d-3	5.7d-3
	4.6d-3	2.9d-3	1.5d-3	7.6d-4	1.8d-4	4.6d-5
	0.0d+0	0.0d+0				
0.0d+0	1.3d-5	5.0d-5	1.5d-4	3.7d-4	4.1d-4	
	1.4d-3	2.2d-3	3.2d-3	4.0d-3	4.1d-3	3.3d-3
	2.1d-3	1.0d-3	5.4d-4	1.3d-4	3.5d-5	1.1d-5
	0.0d+0	0.0d+0				
0.0d+0	0.0d+0	3.7d-5	1.1d-4	2.7d-4	2.8d-4	
	9.7d-4	1.6d-3	2.3d-3	2.8d-3	2.8d-3	2.3d-3
	1.4d-3	7.2d-4	2.0d-4	9.2d-5	2.9d-5	1.2d-5
	0.0d+0	0.0d+0				
1.0d-5	3.3d-5	8.7d-5	1.9d-4	3.5d-4	5.5d-4	
	8.0d-4	1.1d-3	1.4d-3	1.6d-3	1.6d-3	1.3d-3
	9.3d-4	5.5d-4	2.7d-4	1.1d-4	4.0d-5	1.2d-5
	0.0d+0	0.0d+0				
1.0d-5	3.1d-5	8.4d-5	1.9d-4	3.5d-4	5.9d-4	
	9.3d-4	1.4d-3	1.9d-3	2.3d-3	2.3d-3	2.0d-3
	1.5d-3	8.8d-4	4.5d-4	2.0d-4	7.3d-5	2.3d-5
	0.0d+0	0.0d+0				
1.0d-5	3.2d-5	8.8d-5	2.0d-4	3.9d-4	7.1d-4	
	1.2d-3	1.9d-3	2.7d-3	3.3d-3	3.5d-3	3.0d-3
	2.2d-3	1.4d-3	7.1d-4	3.2d-4	1.2d-4	4.0d-5
	1.1d-5	0.0d+0				
1.2d-5	3.8d-5	1.1d-4	2.5d-4	5.1d-4	9.5d-4	
	1.7d-3	2.7d-3	4.0d-3	4.9d-3	5.2d-3	4.5d-3
	3.3d-3	2.0d-3	1.1d-3	4.8d-4	1.8d-4	6.1d-5
	1.7d-5	0.0d+0				
1.6d-5	5.1d-5	1.4d-4	3.3d-4	7.0d-4	1.4d-3	
	2.5d-3	4.1d-3	5.9d-3	7.3d-3	7.5d-3	6.5d-3
	4.7d-3	2.9d-3	1.5d-3	6.6d-4	2.6d-4	8.4d-5
	2.4d-5	0.0d+0				
2.1d-5	6.8d-5	1.9d-4	4.4d-4	9.5d-4	1.9d-3	
	3.5d-3	5.8d-3	8.4d-3	1.0d-2	1.1d-2	9.0d-3
	6.4d-3	3.8d-3	1.9d-3	8.5d-4	3.2d-4	1.1d-4
	3.0d-5	0.0d+0				
2.8d-5	8.4d-5	2.3d-4	5.4d-4	1.2d-3	2.4d-3	
	4.6d-3	7.7d-3	1.1d-2	1.4d-2	1.4d-2	1.2d-2
	8.1d-3	4.8d-3	2.4d-3	1.0d-3	3.8d-4	1.2d-4
	3.4d-5	0.0d+0				
3.8d-5	1.0d-4	2.5d-4	6.0d-4	1.3d-3	2.8d-3	
	5.4d-3	9.2d-3	1.3d-2	1.6d-2	1.7d-2	1.4d-2
	9.6d-3	5.5d-3	2.7d-3	1.1d-3	4.2d-4	1.3d-4
	3.7d-5	0.0d+0				

5.5d-5	1.2d-4	2.7d-4	6.1d-4	1.4d-3	2.9d-3	
5.8d-3	1.0d-2	1.5d-2	1.8d-2	1.8d-2	1.5d-2	
1.0d-2	5.9d-3	2.8d-3	1.2d-3	4.2d-4	1.4d-4	
3.9d-5	1.1d-5					
8.5d-5	1.6d-4	2.9d-4	5.9d-4	1.3d-3	2.8d-3	
5.5d-3	9.7d-3	1.4d-2	1.7d-2	1.8d-2	1.5d-2	
1.0d-2	5.7d-3	2.7d-3	1.1d-3	4.1d-4	1.4d-4	
4.5d-5	1.5d-5					
1.3d-4	2.1d-4	3.3d-4	5.6d-4	1.1d-3	2.4d-3	
4.8d-3	8.4d-3	1.2d-2	1.5d-2	1.5d-2	1.3d-2	
8.7d-3	5.0d-3	2.4d-3	9.9d-4	3.8d-4	1.4d-4	
6.0d-5	2.7d-5					
1.8d-4	2.8d-4	3.8d-4	5.3d-4	9.1d-4	1.9d-3	
3.7d-3	6.6d-3	9.9d-3	1.2d-2	1.2d-2	1.0d-2	
7.0d-3	4.0d-3	1.9d-3	8.3d-4	3.5d-4	1.7d-4	
9.0d-5	4.8d-5					
2.1d-4	3.2d-4	4.2d-4	5.1d-4	7.4d-4	1.4d-3	
2.7d-3	4.8d-3	7.1d-3	8.8d-3	8.9d-3	7.4d-3	
5.1d-3	2.9d-3	1.4d-3	6.8d-4	3.5d-4	2.2d-4	
1.4d-4	8.2d-5					
2.2d-4	3.4d-4	4.2d-4	4.7d-4	5.8d-4	9.7d-4	
1.8d-3	3.2d-3	4.8d-3	6.0d-3	6.0d-3	5.0d-3	
3.5d-3	2.0d-3	1.0d-3	5.7d-4	3.8d-4	2.9d-4	
2.0d-4	1.2d-4					
2.0d-4	3.0d-4	3.7d-4	4.0d-4	4.4d-4	6.6d-4	
1.2d-3	2.1d-3	3.1d-3	3.9d-3	3.9d-3	3.3d-3	
2.2d-3	1.3d-3	7.5d-4	5.0d-4	4.1d-4	3.5d-4	
2.6d-4	1.6d-4					
1.6d-4	2.3d-4	2.9d-4	3.0d-4	3.2d-4	4.4d-4	
7.9d-4	1.4d-3	2.0d-3	2.5d-3	2.5d-3	2.1d-3	
1.4d-3	8.6d-4	5.5d-4	4.4d-4	4.3d-4	3.9d-4	
3.0d-4	1.8d-4					
1.0d-4	1.6d-4	1.9d-4	2.0d-4	2.1d-4	2.9d-4	
5.2d-4	9.0d-4	1.3d-3	1.6d-3	1.6d-3	1.3d-3	
8.9d-4	5.6d-4	4.0d-4	3.8d-4	3.9d-4	3.7d-4	
2.8d-4	1.8d-4					
5.9d-5	8.9d-5	1.1d-4	1.1d-4	1.3d-4	1.9d-4	
3.6d-4	6.2d-4	9.1d-4	1.1d-3	1.1d-3	8.4d-4	
5.6d-4	3.6d-4	2.8d-4	2.9d-4	3.2d-4	3.0d-4	
2.3d-4	1.5d-4					
2.9d-5	4.3d-5	5.3d-5	5.9d-5	7.5d-5	1.3d-4	
2.5d-4	4.4d-4	6.4d-4	7.4d-4	7.1d-4	5.5d-4	
3.6d-4	2.3d-4	1.8d-4	2.0d-4	2.2d-4	2.1d-4	
1.7d-4	1.0d-4					
1.2d-5	1.8d-5	2.3d-5	2.8d-5	4.3d-5	8.7d-5	
1.8d-4	3.1d-4	4.5d-4	5.2d-4	4.8d-4	3.7d-4	
2.3d-4	1.4d-4	1.1d-4	1.2d-4	1.4d-4	1.3d-4	
1.0d-4	6.2d-5					
1.7d-4	1.7d-4	1.3d-4	9.8d-5	1.3d-4	3.5d-4	
9.4d-4	2.1d-3	3.6d-3	4.9d-3	5.4d-3	4.9d-3	
3.7d-3	2.4d-3	1.3d-3	6.3d-4	2.6d-4	8.6d-5	
2.3d-5	0.0d+0					
2.0d-4	2.0d-4	1.6d-4	1.1d-4	1.5d-4	3.9d-4	
1.0d-3	2.3d-3	3.9d-3	5.4d-3	6.0d-3	5.4d-3	
4.1d-3	2.6d-3	1.5d-3	7.0d-4	2.9d-4	9.6d-5	
2.6d-5	0.0d+0					
2.3d-4	2.3d-4	1.8d-4	1.3d-4	1.6d-4	4.3d-4	
1.1d-3	2.5d-3	4.3d-3	5.9d-3	6.5d-3	5.9d-3	

	4.4d-3	2.9d-3	1.6d-3	7.7d-4	3.1d-4	1.1d-4
	2.9d-5	0.0d+0				
2.6d-4	2.6d-4	2.0d-4	1.5d-4	1.8d-4	4.6d-4	
	1.2d-3	2.7d-3	4.6d-3	6.3d-3	7.0d-3	6.3d-3
	4.8d-3	3.1d-3	1.7d-3	8.3d-4	3.4d-4	1.2d-4
	3.2d-5	0.0d+0				
2.8d-4	2.9d-4	2.3d-4	1.6d-4	1.9d-4	4.9d-4	
	1.3d-3	2.8d-3	4.9d-3	6.7d-3	7.4d-3	6.7d-3
	5.0d-3	3.2d-3	1.8d-3	8.9d-4	3.7d-4	1.3d-4
	3.5d-5	0.0d+0				
3.1d-4	3.1d-4	2.5d-4	1.8d-4	2.1d-4	5.1d-4	
	1.4d-3	3.0d-3	5.1d-3	7.0d-3	7.7d-3	6.9d-3
	5.2d-3	3.4d-3	1.9d-3	9.3d-4	3.9d-4	1.3d-4
	3.7d-5	0.0d+0				
3.3d-4	3.3d-4	2.6d-4	1.9d-4	2.1d-4	5.3d-4	
	1.4d-3	3.0d-3	5.2d-3	7.2d-3	7.9d-3	7.1d-3
	5.4d-3	3.5d-3	2.0d-3	9.6d-4	4.0d-4	1.4d-4
	3.8d-5	0.0d+0				
3.5d-4	3.5d-4	2.8d-4	2.0d-4	2.2d-4	5.4d-4	
	1.4d-3	3.1d-3	5.3d-3	7.2d-3	8.0d-3	7.2d-3
	5.4d-3	3.5d-3	2.0d-3	9.8d-4	4.1d-4	1.4d-4
	4.0d-5	0.0d+0				
3.6d-4	3.7d-4	2.9d-4	2.0d-4	2.2d-4	5.4d-4	
	1.4d-3	3.1d-3	5.3d-3	7.2d-3	7.9d-3	7.1d-3
	5.4d-3	3.5d-3	2.0d-3	9.8d-4	4.1d-4	1.4d-4
	4.0d-5	0.0d+0				
3.7d-4	3.7d-4	2.9d-4	2.0d-4	2.2d-4	5.3d-4	
	1.4d-3	3.0d-3	5.2d-3	7.0d-3	7.7d-3	7.0d-3
	5.3d-3	3.4d-3	2.0d-3	9.7d-4	4.1d-4	1.4d-4
	4.0d-5	0.0d+0				
3.7d-4	3.7d-4	2.9d-4	2.0d-4	2.2d-4	5.1d-4	
	1.3d-3	2.9d-3	5.0d-3	6.8d-3	7.4d-3	6.7d-3
	5.1d-3	3.3d-3	1.9d-3	9.5d-4	4.0d-4	1.4d-4
	3.9d-5	0.0d+0				
3.6d-4	3.7d-4	2.9d-4	2.0d-4	2.1d-4	4.8d-4	
	1.3d-3	2.7d-3	4.7d-3	6.4d-3	7.1d-3	6.4d-3
	4.8d-3	3.2d-3	1.8d-3	9.1d-4	3.9d-4	1.4d-4
	3.8d-5	0.0d+0				
3.5d-4	3.5d-4	2.8d-4	1.9d-4	2.0d-4	4.5d-4	
	1.2d-3	2.5d-3	4.4d-3	6.0d-3	6.6d-3	5.9d-3
	4.5d-3	3.0d-3	1.7d-3	8.6d-4	3.7d-4	1.3d-4
	3.6d-5	0.0d+0				
3.3d-4	3.3d-4	2.6d-4	1.8d-4	1.8d-4	4.1d-4	
	1.1d-3	2.3d-3	4.0d-3	5.5d-3	6.0d-3	5.4d-3
	4.1d-3	2.7d-3	1.6d-3	8.0d-4	3.4d-4	1.2d-4
	3.4d-5	0.0d+0				
3.1d-4	3.1d-4	2.4d-4	1.7d-4	1.7d-4	3.7d-4	
	9.7d-4	2.1d-3	3.6d-3	4.9d-3	5.4d-3	4.9d-3
	3.7d-3	2.5d-3	1.5d-3	7.4d-4	3.2d-4	1.1d-4
	3.2d-5	0.0d+0				
2.8d-4	2.9d-4	2.2d-4	1.5d-4	1.5d-4	3.3d-4	
	8.6d-4	1.9d-3	3.2d-3	4.4d-3	4.8d-3	4.4d-3
	3.3d-3	2.2d-3	1.3d-3	6.7d-4	2.9d-4	1.0d-4
	2.9d-5	0.0d+0				
2.6d-4	2.6d-4	2.0d-4	1.4d-4	1.3d-4	2.9d-4	
	7.5d-4	1.6d-3	2.8d-3	3.8d-3	4.2d-3	3.8d-3
	2.9d-3	2.0d-3	1.2d-3	5.9d-4	2.6d-4	9.1d-5
	2.6d-5	0.0d+0				

2.3d-4	2.3d-4	1.8d-4	1.2d-4	1.1d-4	2.5d-4	
6.4d-4	1.4d-3	2.4d-3	3.2d-3	3.6d-3	3.3d-3	
2.5d-3	1.7d-3	1.0d-3	5.2d-4	2.3d-4	8.1d-5	
2.3d-5	0.0d+0					
2.0d-4	2.0d-4	1.5d-4	1.0d-4	9.7d-5	2.1d-4	
5.4d-4	1.2d-3	2.0d-3	2.7d-3	3.0d-3	2.8d-3	
2.1d-3	1.5d-3	8.7d-4	4.5d-4	2.0d-4	7.0d-5	
2.0d-5	0.0d+0					
1.7d-4	1.7d-4	1.3d-4	8.7d-5	8.1d-5	1.7d-4	
4.4d-4	9.6d-4	1.7d-3	2.3d-3	2.5d-3	2.3d-3	
1.8d-3	1.2d-3	7.4d-4	3.8d-4	1.7d-4	6.0d-5	
1.7d-5	0.0d+0					

50.47d0 10800.d0 1000.d0
 6.32d0 1207.d0 1000.d0
 9.48d0 2245.d0 1000.d0
 17.37d0 4439.d0 1000.d0
 5.81d0 1440.d0 1000.d0
 10.d0 2903.d0 1000.d0
 160.d0 16.29d0 24.49d0 1000.d0
 36.15d0 7.11d0 66.70d0 1000.d0
 4.4d0 1.0d0
 4.9d0 1.4d0
 3.8d0 0.9d0
 4.2d0 1.1d0
 -0.958d0 1.61d0
 -1.27d0 1.66d0
 0.42d0 2.76d0 0.55d0 1.24d0
 0.7824d0 0.4928d0 0.5889d0 0.6491d0
 -0.2524d0 1.046d0
 -0.06388d0 1.842d0
 0.2449d0 0.5327d0
 0.09721d0 1.107d0
 18.d0 7.668d0
 9.d0 3.420d0
 6.d0 2.016d0
 3.d0 0.894d0
 2.d0 0.738d0
 30.d0 0.063d0
 60.d0 0.297d0
 90.d0 0.384d0
 120.d0 0.142d0
 150.d0 0.114d0
 2.9546d0 0.d0 1.513d0
 2000.d0 4000.d0 7000.d0 13000.d0 4000.d0 4000.d0
 0.d0 0.d0 0.d0 0.d0 0.d0
 0.d0 0.d0 0.d0 0.d0 0.d0
 5.074d-7 1.106d-6 0.58d0 0.74d0
 2.115d-6 4.797d-6 0.58d0 0.74d0
 9.723d-7 3.203d-6 0.59d0 0.50d0
 2.430d-6 6.628d-6 0.75d0 0.50d0
 1.426d-6 3.422d-6 0.60d0 0.83d0
 1.211d-5 2.100d-5 0.50d0 0.50d0
 9.797d-6 2.294d-5 0.50d0 0.50d0
 3.893d-6 8.830d-6 0.50d0 0.50d0
 1.481d-6 5.117d-6 0.50d0 0.50d0
 1.053d-5 1.341d-5 0.50d0 0.50d0
 3.310d-6 4.264d-6 0.50d0 0.50d0

4.4. The output file, "crout".

The output file "crout" summarizes the input preferences in "crin" and displays requested probabilities. Recall that in "crin" the user specifies the number of critical velocities, runways, and structures to be analyzed. Any of these numbers may be unity, and a critical velocity may be zero. The probabilities displayed are

- for each requested (critical velocity, runway, structure) combination the probability (rate) that for a single operation using the runway in question the given structure would be hit by an aircraft at a velocity exceeding the given critical velocity
- for each requested (critical velocity, runway, structure) combination the probability that during the specified time window the given structure would be hit by an aircraft using the runway in question at a velocity exceeding the given critical velocity
- for each requested (critical velocity, runway) combination the probability that during the specified time window any structure would be hit by an aircraft using the runway in question at a velocity exceeding the given critical velocity
- for each requested (critical velocity, structure) combination the probability that during the specified time window the given structure would be hit by an aircraft using any runway at a velocity exceeding the given critical velocity
- for each requested critical velocity the probability that during the specified time window any structure would be hit by an aircraft using any runway at a velocity exceeding the given critical velocity.