A STUDY OF

TURBULENT FLAME PROPAGATION

by

Dinah Georgianna McNutt

B.S.Eng. University of Tennessee
(1979)

SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

DECEMEBER 1981
(i.e. Feb '82)

Signature of Author Signature redacted
Department of Mechanical Engnineering

Certified by Signature redacted
Thesis Supervisor

Signature redacted

Accepted by
Chairman, Departmental Committee on Graduate Students

A STUDY OF TURBULENT FLAME PROPAGATION

by

Dinah Georgianna McNutt

Submitted to the Department of Mechanical Engineering
on December 21, 1981 in partial fulfillment of the
requirements for the Degree of Master of Science
in Mechanical Enigneering.

This thesis examines existing turbulence theories through numerical methods. Two different methods are used to model a premixed turbulent flame: a moment model and a Monte Carlo model.

The moment model solves the transport equation for the nondimensional progress variable, C, and assumes small fluctuations of C about the mean. The Monte Carlo model solves the probability density function (pdf) equation of C using statistical methods. The flame is described by a nondimensional grid composed of ensembles of elements. The Monte Carlo model assumes the steepest scalar gradients are due to scalar dissipation caused by turbulent straining. Mixing is modeled by the Curl mixing model and reaction is simulated by integrating the nondimensional rate equation. Both models provide information about flame speed, flame thickness, and profiles of C.

Numerical calculations from the moment model show that the moment model is valid for only a limited range of the operating conditions. From the pdf calculations of the Monte Carlo model it is shown that the basic assumptions behind the moment model are invalid. The results from the Monte Carlo calculations show good agreement with the theory. Pdf calculations show that the model is mixing limited. Results of calculated values of flame speed and flame thickness show that the assumption of gradient diffusion is invalid for the case of a laminar flame.

Thesis Supervisor: Stephen B. Pope
Title: Associate Professor of Mechanical Engineering

Table of Contents

NOMENCLATURE

C	nondimensional product concentration
c'	fluctuations of C about the mean
C_μ , C_ϕ	turbulence model constants
$E(\psi)$	mixing term, Equation (2.2)
J	number of ensembles
K	turbulent kinetic energy
ℓ_ℓ ℓ_ℓ^*	laminar flame thickness, normalized laminar flame thickness
ℓ_T ℓ_T^*	turbulent flame thickness, normalized turbulent flame thickness
ℓ_ϵ	integral macroscale
N	number of elements in an ensemble
N_d	number of elements involved in diffusion process
N_m	number of elements involved in mixing process
$P(\psi)$	pdf of ϕ
R	turbulent Reynolds number
$S(C)$, $S^*(C)$	reaction rate, normalized reaction rate
$S(\phi)$	rate of change of ϕ due to chemcial reaction
S_ℓ , S_ℓ^*	laminar flame speed, normalized laminar flame speed
S_T , S_T^*	turbulent flame speed, normalized turbulent flame speed
t, t^*	time, nondimensional time
U, U^*	velocity, normalized velocity
u'	turbulent intensity, fluctutation of U about the mean
X, X^*	position, nondimensional position

Greek Letters

Γ_T	turbulent diffusion coefficient
η	Kolomogorov length scale
λ	Taylor microscale
υ	transport coefficient
υ_{eff}	effective transport coefficient
υ_T	turbulent transport coefficient
σ	standard deviation
ϕ	scalar variable
ψ	independent space variable corresponding to ϕ
Ω	Damkohler number
ω	turbulent frequency
τ	Kolmogorov time scale
τ_R	chemical reaction rate

Subscritps and Superscripts

D	refers to Damkohler theory
i	coordinate direction i
ℓ	laminar
M	refers to Moment model
MC	refers to Monte Carlo model
T	turbulent

Averages

$\langle S \rangle$	mean value
$S^\bullet$	normalized value

I would like to thank Professor Stephen Pope for his technical guidance during this project. I would also like to acknowledge International Harvester for their support.

A big thanks goes to Jim Watts for keeping the PDP-11 running and to everyone in the Slaon Automotive Lab for their help and moral support. Thanks to Tony for all of the advice on doing figures, etc. and to Rick and the rest of the guys in my office who made the working atmosphere pleasant. I especially want to thank Suzy Leyes for letting me use her car,office,keys, etc. I also want to thank Mark Nagurka for the daily jokes and the rest of the crew in the VDL.

A big debt of graditude goes to my folks for their love and support.

This work was sponsored by a fellowship from International Harvester.

INTRODUCTION

Turbulent flames occur in power plants, industry, and transportation. The properties of a turbulent flame affect the efficiency of the process involved. For example, in a spark-ignition engine, the turbulent flame speed and thickness are two of the parameters used as empirical inputs in analyzing engine performance. If the flame speed and flame thickness could be obtained theoretically, the engine simulations could be performed for a wide range of operating conditions without having to make actual experimental measurements.

The purpose of this research project is to develop a theory to determine quantitively the flame speed and flame thickness of premixed turbulent flames. The motivation for this project arose from inconsistencies in the literature, dimensionally incorrect relationships, and conflicting assumptions in turbulence models. One objective of this study is to identify important dimensionless parameters which define operating conditions of a turbulent flame and to determine the normalized flame speed and thickness as a function of these parameters. Another goal is to identify different regimes of combustion and the limits of these regimes. Two turbulence models are examined and the results of the models are compared to theory.

CHAPTER 1

The simple case of a one-step irreversible reaction between a fuel and an oxidant to form a single product is considered. Initially, the fuel and oxidant are homogeneously premixed. The density is assumed to be constant and the transport properties of all the species and enthalpy are assumed to be equal and constant. The chemistry can be described by a single scalar variable which can be chosen to be the progress variable C. This variable may be thought of as the nondimensional product concentration: in the unburnt mixture C is equal to zero, after complete combustion C is equal to unity.

The conservation equation for C(X,t) is

$$\frac{\partial C}{\partial t} + U_i \frac{\partial C}{\partial X_i} = \gamma \frac{\partial^2 C}{\partial X_i^2} + S(C) \qquad (1.1)$$

where $\underline{U}(X,t)$ is the velocity at location $\underline{X}$ and time t, γ is the transport coefficient, and S(C) is the source term due to chemical reaction. The source S(C) is zero for C=0 and C=1 since neither cold, pure reactants nor fully burnt products can react. Since the reaction is irreversible, S(C) is non-negative. A characteristic chemical time scale, τ_R , is defined by

$$\frac{1}{\tau_R} = \text{Sup} \left| S(C) \right| \qquad (1.2)$$

and the normalized reaction rate is defined by

$$S^*(C) = \tau_R S(C). \tag{1.3}$$

Thus, $S^*(C)$ is a nondimensional source term that varies between zero and unity.

In a quiescent fluid, Equation (1.1) admits a solution corresponding to a plane laminar flame. If the flame propagates at a speed S_ℓ (the laminar flame speed), then Equation (1.1) written in a frame moving with the flame becomes

$$S_\ell \frac{dc}{dx} = \gamma \frac{d^2c}{dx^2} + S(C). \tag{1.4}$$

This equation can be nondimensionalized using the known parameters γ and τ_R. Therefore, defining

$$S_\ell^* \equiv S_\ell (\tau_R / \gamma)^{1/2} \tag{1.5}$$

and

$$X^* \equiv X(\tau_R \gamma)^{-1/2}, \tag{1.6}$$

Equation (1.4) can now be written as

$$S_\ell^* \frac{dC}{dX^*} = \frac{d^2C}{dX^{*2}} + S^*(C). \tag{1.7}$$

Figure 1 shows a graphical representation of the solution to this equation. S_ℓ^* is the nondimensional flame speed and ℓ_ℓ^* is the nondimensional flame thickness defined by

$$\ell_\ell^* = X_{0.9}^* - X_{0.1}^* \tag{1.8}$$

where

$$C(X^*_{0.9}) = 0.9 \tag{1.9}$$

and

$$C(X^*_{0.1}) = 0.1 \tag{1.10}$$

as shown in Figure 1. These nondimensional flame properties, S^*_ℓ and ℓ^*_ℓ, depend upon $S^*(c)$ and are of order unity. For different values of γ and τ_R, the dimensional quantities S_ℓ and ℓ_ℓ can be recovered from

$$S_\ell = (\gamma / \tau_R)^{1/2} S^*_\ell \tag{1.11}$$

and

$$\ell_\ell = (\gamma \tau_R)^{1/2} \ell^*_\ell . \tag{1.12}$$

In all of the calculations reported here, the normalized reaction rate was taken to be[1]

$$S^*(C) = 6.11 \times 10^7 C(1-C) \exp\left[-30{,}000/(300 + 600C)\right] \tag{1.13}$$

For this reaction rate, numerical solution of Equation (1.7) (see Chapter 3) yields

$$S^*_\ell = 0.768 \tag{1.14}$$

and

$$\ell^*_\ell = 4.780 . \tag{1.15}$$

These values are used in all subsequent calculations and figures.

For the artificial case of homogeneous, nondecaying turbulence, the combustion of a turbulent mixture of fuel and oxidant produces a turbulent flame. The properties of the flame are determined by several parameters: u', ℓ_ϵ, τ_R, and ν. The turbulent intensity, u', is defined as the square root of the turbulent kinetic energy, K. The integral macroscale, ℓ_ϵ, is the characteristic size of the largest eddies and is defined so that

$$\ell_\epsilon \equiv K^{3/2} / \epsilon \tag{1.16}$$

where ϵ is the rate of dissipation of the turbulent kinetic energy. Molecular transport is described by the kinematic viscosity, ν.

From the four quantities, u', ℓ_ϵ, ν, and τ_R, two independent, dimensionless groups can be obtained. The choice is not unique: here the two groups are chosen to be

$$R = u' \ell_\epsilon / \nu \tag{1.17}$$

and

$$\Omega = \ell_\epsilon / (u' \tau_R). \tag{1.18}$$

R is the turbulent Reynolds number and Ω is the ratio of

chemical frequency to turbulent frequency or the Damkohler number. All other dimensionless groups can be expressed in terms of R and Ω . For example,

$$\frac{S_\ell}{u'} = \frac{(\gamma / \tau_R)^{1/2}}{u'} S_\ell^* = \left(\frac{\Omega}{R}\right)^{1/2} S_\ell^* \quad . \tag{1.19}$$

and

$$\frac{\ell_\ell}{\ell_\epsilon} = \frac{(\gamma \ \tau_R)^{1/2}}{\ell_\epsilon} \ell_\ell^* = \left(\frac{1}{R\,\Omega}\right)^{1/2} \ell_\ell^* \ . \tag{1.20}$$

Any flame can be described in terms of R and Ω only, and each point on the R-Ω plane corresponds to a set of operating conditions for a flame. Figure 2 shows contours of S_ℓ / u' and $\ell_\ell / \ell_\epsilon$ plotted on the R-Ω plane. In the region of the plane where $\ell_\ell >> \ell_\epsilon$, the turbulent eddies are too small to affect the flame structure, but they augment the transport process in the flame. This transport process governs the flame speed by increasing the transfer of fluid between the preheat zone and the reaction zone in the flame.[2] The effective transport coefficient, γ_{eff} , can be defined by

$$\gamma_{eff} = \gamma + \gamma_T \tag{1.21}$$

where γ_T is the turbulent viscosity. An expression for γ_T is obtained from the k-ϵ turbulence model.[3]

$$\gamma_T = C_\mu u' \ell_\epsilon \tag{1.22}$$

where the constant C_μ is ascribed the value 0.09.

Since the only effect on the flame is the increase in the transport coefficient, the same laminar equations apply (e.g. Equations (1.19) and (1.2)), but with γ_{eff} replacing γ. Noting that both S_ℓ and ℓ_ℓ are proportional to $\gamma^{1/2}$, expressions for S_T/S_ℓ and ℓ_T/ℓ_ℓ are written as

$$\frac{S_T}{S_\ell} = \left(\frac{\gamma_{eff}}{\gamma}\right)^{1/2} \tag{1.23}$$

and

$$\frac{\ell_T}{\ell_\ell} = \left(\frac{\gamma_{eff}}{\gamma}\right)^{1/2}. \tag{1.24}$$

Substituting Equations (1.11) and (1.12) for S_ℓ and ℓ_ℓ, the turbulent flame speed and the turbulent flame thickness may be written as

$$S_T = (\gamma_{eff}/\tau_R)^{1/2} S_\ell^* \tag{1.25}$$

and

$$\ell_T = (\gamma_{eff}\,\tau_R)^{1/2} \ell_\ell^*. \tag{1.26}$$

Using the definitions for γ_{eff} and γ_T as given in Equations (1.21) and (1.22), the expressions for S_T and ℓ_T can be rewritten as

$$S_T = \left(\frac{\gamma + C_\mu u' \ell_\epsilon}{\tau_R}\right)^{1/2} S_\ell^* \tag{1.27}$$

and

$$\ell_T = ((\gamma + C_\mu u' \ell_\epsilon)\tau_R)^{1/2} \ell_\ell^* . \tag{1.28}$$

Normalizing S_T and ℓ_T with the turbulent intensity and the integral macroscale, respectively, and substituting for R and Ω, Equations (1.27) and (1.28) become

$$\frac{S_T}{u'} = \left(\frac{\Omega}{R} + C_\mu \Omega\right)^{1/2} S_\ell^* \tag{1.29}$$

and

$$\frac{\ell_T}{\ell_\epsilon} = \left(\frac{1}{R\,\Omega} + \frac{C_\mu}{\Omega}\right)^{1/2} \ell_\ell^* . \tag{1.30}$$

Figures 3 and 4 show contours of S_T/u' and ℓ_T/ℓ_ϵ using the values in Equations (1.14) and (1.15) for S_ℓ^* and ℓ_ℓ^*. For large Reynolds numbers, these contours become independent of R.

In fact, as long as the turbulent flame thickness is much greater than the integral macroscale, this theory still applies, since the transport process still governs the flame speed and the turbulent eddies are still too small to affect the flame structure. As R approaches infinity ℓ_T/ℓ_ϵ becomes inversely proportional to the square root of the Damkohler number. Thus, the Damkohler theory is valid below some value of Ω. The calculations in Chapter 3 suggest this value is

Ω =0.1.

The ratio of turbulent transport, γ_{eff} , to molecular transport, γ , can be expressed in terms of the Reynolds number only. Using Equations (1.21), (1.22), and (1.17), this ratio becomes

$$\frac{\gamma_{eff}}{\gamma} = 1 + \frac{C_\mu u' \ell_\epsilon}{\gamma} = 1 + C_\mu R . \qquad (1.31)$$

Thus, at least at high Reynolds number, turbulent transport dominates molecular transport for values of the Damkohler number less than 0.1. The ratio of turbulent straining to laminar straining is

$$\frac{(strain)_T}{(strain)_\ell} = \frac{\tau^{-1}}{S_\ell / \ell_\ell} \qquad (1.32)$$

where τ is the Kolmogorov time scale:

$$\tau = (\gamma / \epsilon)^{1/2} \qquad (1.33)$$

Equation (1.32) written in terms of R and Ω becomes

$$\frac{(strain)_T}{(strain)_\ell} = \left(\frac{R}{\Omega^2}\right)^{1/2} \frac{\ell_\ell^*}{S_\ell^*} , \qquad (1.34)$$

Since $\Omega < 0.1$, this expression is greater than unity and the turbulent strain rate is greater than the laminar strain rate. The laminar flame speed and the laminar flame thickness are directly affected by the reaction rate, τ_R , as seen in

equations (1.11) and (1.12).

At the other extreme, if the laminar flame speed is much greater than the turbulent intensity, then the laminar flame front propagates very quickly through the turbulent fluid. The ratio of the laminar time scale to the turbulent time scale is expressed in Equation (1.32) which is less than unity in this region of the $R-\Omega$ plane. Therefore, the laminar time scale is much smaller than the turbulent time scale and the turbulence has little effect on the flame. This region of the $R-\Omega$ plane (see Figure 5) represents a laminar flame. Transport is due to molecular diffusion and the steepest gradients are caused by the reaction zone in the laminar flame.

The behavior of flames in the region of the $R-\Omega$ plane between the Damkohler theory region ($\Omega < 0.1$) and the plane laminar flame region ($S_\ell > u'$) is less certain. This region can be subdivided by the line $\ell_\ell = \eta$, where η is the Kolmogorov length scale

$$\eta \equiv (\nu^3 / \epsilon)^{1/4} . \tag{1.35}$$

The Kolmogorov scale η is the characteristic length scale of the steepest scalar gradients. Figure 6 shows these regions as well as the Damkohler theory region and the laminar flame region.

Region I is characterized by Damkohler numbers greater than 0.1 and $\ell_\ell > \eta$. Since u' >> S_ℓ and $\ell_\ell > \eta$, both transport and mixing are dominated by the turbulence. Also, for most of the region $\Omega >> 1.0$, the rate of combustion is limited by turbulent mixing rather than by the reaction rate. Since both transport and mixing are governed by the turbulence, it is reasonable to assume S_T /u' and ℓ_T/ℓ_ϵ to be independent of Ω and R (for high Reynolds numbers). In this region of the R-Ω plane the eddy-break-up model appears to be applicable.

In Region II, which is characterized by $\ell_\ell < \eta$ and S_ℓ <<u', turbulent transport still dominates molecular transport since the turbulent intensity is much greater than the laminar flame speed. The steepest gradients can now be attributed to laminar flamlets since the laminar flame thickness is less than the Kolmogorov length scale. This effect on the flame properties S_T /u' and ℓ_T/ℓ_ϵ is not clear.

For the purposes of this study the Reynolds number is chosen to be very high (10^6). Therefore, for values of the Damkohler number less than 0.1, S_T /u' and ℓ_T/ℓ_ϵ are independent of the Reynolds number. Figure 7 shows contours of known dimensional parameters plotted in terms of Ω only. The expression ℓ_ℓ/λ is the ratio of laminar flame speed to the Taylor microscale. As Ω approaches infinity, S_ℓ /u'

increases and the other ratios tend to zero. For small values of Ω , Damkohler's theory applies. As Ω is increased, S_T/u' becomes independent of Ω and for large values of the Damkohler number a fast laminar flame exists. Figure 8 shows S_T/u' for these three theories. Since in the intermediate region S_T/u' is independent of Ω , the magnitude of S_T/u' is not known, but the result will be a horizontal line. The actual S_T/u' for a turbulent flame will follow the asymptote for Damkohler's theory, become independent of Ω and then pick up the result for a laminar flame as Ω varies from small to large values. As shown in Chapter 3, the asymptote of Regions I and II is actually not reached until $\Omega = 10^4$. The reasons behind the behavior of this transition are discussed in Chapter 3.

CHAPTER 2

This chapter contains the description of the two models studied. The first model uses a Monte Carlo method to solve the pdf equation of the scalar variable, C. The second model, the moment model, solves Equation (1.7) and assumes small scalar fluctuations of C about the mean, $\langle C \rangle$. Both methods assume homogeneous, nondecaying turbulence in a fluid which, upon combustion, produces a plane propagating flame. Also, both models are independent of Reynolds number and are valid for a limited range of the Damkohler number. The moment model is valid where the Damkohler theory is applicable and also for slightly larger values of Ω . The Monte Carlo model is valid for the Damkohler theory regime and Regions I and II of the R-Ω plane. Values of Ω beyond which these models fail are discussed in Chapter 3. In the limit of small fluctutations (i.e. $\langle C'^4 \rangle \ll 1.0$), the pdf equation reduces to the moment model. In the limit of zero fluctuations (i.e. $\langle C'^2 \rangle \ll 1.0$), both models reduce to Damkohler's theory.

Monte Carlo Model

The Monte Carlo method[6] solves the pdf equation for the scalar variable, C, which is denoted by ϕ and the pdf of ϕ is $P(\psi;X,t)$ where ψ is the one-dimensional composition space variable corresponding to ϕ .[4] The

expression $P(\psi;X,t)$ can be simplified as $P(\psi)$. The one-dimensional transport equation for $P(\psi)$ assumes turbulent transport is modeled by simple gradient diffusion and can be written as

$$\frac{\partial P(\psi)}{\partial t} + \langle U\rangle \frac{\partial P(\psi)}{\partial X} + \frac{\partial}{\partial \psi}\left(P(\psi)S(\psi)\right) = \frac{\partial}{\partial X}\,\Gamma_T\,\frac{\partial P(\psi)}{\partial X} + E(\psi;X,t) \qquad (2.1)$$

where $\langle U\rangle$ is the mean velocity in the X-direction, $S(\phi)$ is the rate of change of ϕ due to reaction, and $\Gamma_T(X,t)$ is the turbulent diffusion coefficient. The term $E(\psi;X,t)$ represents the effect of molecular mixing. The mixing term and the diffusion term must be modeled while the remainder of the terms are exact. Curl's mixing model is employed to model $E(\psi;X,t)$:[5]

$$E(\psi;X,t) = 2\omega\int P(\psi+\psi')\,P(\psi-\psi')\,d\psi' - \omega P(\psi) \qquad (2.2)$$

where ω is the turbulent mixing frequency and $\psi=\psi'$ is a location in composition space corresponding to fluid with property $\phi=\psi'$. The expressions for Γ_T and ω are obtained from the K-ϵ turbulence model:[3]

$$\Gamma_T = \frac{C_\mu}{\sigma_\phi}\,\frac{K^2}{\epsilon} \qquad (2.3)$$

and

$$\omega = 2C_\phi\frac{\epsilon}{K} \qquad (2.4)$$

where K is the turbulent kinetic energy and the constants C_μ , σ_ϕ , and C_ϕ are assigned the values of 0.09, 0.7, and 2.0, respectively. The normalized diffusion coefficient, Γ_T^* , and the normalized turbulent frequency, ω^*, are defined by

$$\Gamma_T^* \equiv \frac{\Gamma_T}{u' \ell_\epsilon} \tag{2.5}$$

and

$$\omega^* \equiv \frac{\omega}{u' \ell_\epsilon} . \tag{2.6}$$

Using the definitions of u' and ℓ_ϵ , Equations (2.5) and (2.6) give

$$\Gamma_T^* = \frac{C_\mu}{\sigma_\phi} \tag{2.7}$$

and

$$\omega^* = 2C_\phi . \tag{2.8}$$

The model solves equation (2.1) for the case of a statistically stationary, planar flame in homogeneous turbulence. The model uses a one-dimensional finite difference grid of length X^* and composed of J grid nodes spaced ΔX^* distance apart. Each grid node contains an ensemble, or group, of N elements and each element carries the value of the nondimensional product concentration, C. The ensemble average concentration at the jth node, $\langle C \rangle_j$, is

$$\langle C \rangle_j = \frac{1}{N} \sum_{n=1} C_j^n \tag{2.9}$$

where C_i^n is the concentration of the nth element. The boundary conditions for the grid are

$$<C>_1 = 1.0 \tag{2.10}$$

and

$$<C>_J = 0.0 \, . \tag{2.11}$$

The scalar fluctutation from the mean, C' , is

$$C'_j = \sqrt{<C'^2>_j} \tag{2.12}$$

where

$$<C'^2>_j = \frac{1}{N} \sum_{n=1}^{N} (C_j^n - <C>_j)^2 \, . \tag{2.13}$$

The pdf, $P(\psi)$, at a chosen grid node can be determined by creating a histogram of $P(\psi)$ versus ψ from the known scalar values in the ensemble. If ϕ_r represents a random scalar variable, then the probability distribution function, $F(\psi)$, is

$$F(\psi) = \text{Prob} \left| \phi_r < \psi \right| \tag{2.14}$$

and

$$P(\psi) = \frac{dF(\psi)}{d\psi} \, . \tag{2.15}$$

If ψ_a and ψ_b are two independent scalar variables and $\psi_b < \psi_a$, then

$$\text{Prob}\left| \psi_b < \phi_r < \psi_a \right| = F(\psi_a) - F(\psi_b). \quad (2.16)$$

This equation can be applied to form the histogram of $P(\psi)$. The X-axis of the histogram is divided into K divisions of $\Delta\psi$ width. Equation (2.16) now becomes

$$\text{Prob}\left| \psi_b < \phi_r < \psi_b + \Delta\psi \right| = F(\psi_b + \Delta\psi) - F(\psi_b). \quad (2.17)$$

As $\Delta\psi$ approaches zero, the histogram approaches the true shape of the pdf. For each value of C in a given ensemble, the corresponding block in the histogram, K , is determined using

$$k^n = C^n K . \quad (2.18)$$

The pdf is found by normalizing the number of elements in each block with the total number of elements, N. Therefore, $F(\psi)$ varies between zero and unity and the pdf of a fully burned mixture ($\langle C \rangle = 1.0$) is a Dirac delta function at $\psi = K$.

The ensembles are modified at each time step in order to simulate the four processes governing the evolution of $P(\psi)$: convection, reaction, diffusion, and mixing. The convection process is ignored since the coordinate system is chosen so that the mean velocity is zero.

With Δt^* being the normalized time step, then the number of pairs from each ensemble involved in the mixing process at each step, N_m , and the number of elements involved in the

diffusion process, N_d , are

$$N_m = \frac{1}{2} N \omega^* \Delta t^* \tag{2.19}$$

and

$$N_d = \Gamma_T^* \Delta t^* N/(\Delta X^*)^2 \tag{2.20}$$

Mixing occurs by choosing two different elements within an ensemble at random and allowing the elements to mix completely so that they reach a mean scalar concentration. Both elements are assigned a new scalar concentration equal to the average of the two prior to mixing. This process is repeated N_m times for each ensemble at each time step.

Diffusion is simulated by selecting two sets of N_d elements at random from each ensemble and commuting one set forwards (from node j to node j+1) and the other set backwards (from node j to node j-1). This process is repeated for each grid node and the boundary conditions are applied so that there is no loss of elements in each ensemble.

Reaction is simulated deterministically from Equation (1.13):

$$\frac{dC}{dt^*} = S^*(C) \tag{2.21}$$

Equation (2.21) can be rewritten as

$$\frac{dt^*(C)}{dC} = \frac{1}{S^*(C)} \tag{2.22}$$

where $t^*(C)$ is the nondimensional time associated with C. Since C^n at time $t^*(C^n)$ is known, C^n at time $t^*(C^n)+\Delta t_r^*$, where Δt_r^* is the time elapsed since the element last reacted, can be found by integrating equation (2.22) with respect to C using a fourth-order Runge-Kutta scheme. It would be computationally inefficient to perform this integration for every element at each time step. Therefore, the results of the integration scheme are tabulated and the reaction process is updated only when needed (e.g. when mixing occurs or pdf calculations are performed.) For a given value of C^n, the table is searched, using linear interpolation, for a corresponding $t^*(C^n)$. The new concentration after reaction, C^*, is found by locating the corresponding value of C^* for $t^*(C^*)$ which is determined by

$$t^*(C^*) = t^*(C^n) + \Delta t_r^* . \qquad (2.23)$$

Actually, Δt^* for the mixing and diffusion processes are not the same. The Δt^* for each process is chosen so that not more than 10% of the elements from each ensemble are involved in the process at a given time. Therefore, from Equations (2.19) and (2.20),

$$\Delta t_m^* = N_m / (\tfrac{1}{2} N \omega^*) \qquad (2.24)$$

and

$$\Delta t_d^* = N_d (\Delta X^*)^2 / (\Gamma_T^* N) \qquad (2.25)$$

where

$$N_m = (0.05)N \tag{2.26}$$

and

$$N_d = (0.1)\ N \tag{2.27}$$

and Δt_m^* and Δt_d^* are the time steps for the mixing and diffusion processes, respectively.

The flame thickness is measured according to Equation (1.8) with ℓ_T^* replacing ℓ_ℓ^*. The location of $\langle C(X_{0.9}^*)\rangle$ and $\langle C(X_{0.1}^*)\rangle$ are determined using linear interpolation. The normalized turbulent flame speed, which is defined as

$$S_T^* \equiv \frac{S_T}{u'} \tag{2.28}$$

is determined by calculating the speed of the flame at $X_{0.5}^*$. The velocity of the reference frame is known and, therefore, the flame speed is the velocity of the reference frame (velocity of the grid) plus the velocity of the flame with respect to the grid.

Moment Model

The moment model solves the scalar transport equation

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial X_i}\left(U_i C\right) = S(C) + \nu \nabla^2 C \tag{2.29}$$

for $\langle C \rangle$ and $\langle C'^2 \rangle$ where

$$C = \langle C \rangle + C' \tag{2.30}$$

and

$$U_i = \langle U_i \rangle + U_i' \tag{2.31}$$

By taking the mean of Equation (2.29)

$$\frac{\partial \langle C \rangle}{\partial t} + \frac{\partial}{\partial X_i} \langle U_i\, C \rangle = \langle S(C) \rangle + \nu \nabla^2 \langle C \rangle \tag{2.32}$$

and subtracting (2.32) from (2.29) a transport equation for C' is obtained:

$$\frac{\partial C'}{\partial t} + \frac{\partial}{\partial X_i} [\, U_i' \langle C \rangle + U_i C' - \langle U_i'\, C' \rangle \,] = S(C) - \langle S(C) \rangle + \nu \nabla^2 C' \tag{2.33}$$

To derive the one-dimensional transport equation for $\langle C'^2 \rangle$ Equation (2.33) is multiplied by $2C'$ and the mean of the resulting equation is

$$\frac{\partial \langle C'^2 \rangle}{\partial t} + \langle U_i \rangle \frac{\partial \langle C'^2 \rangle}{\partial X_i} = - \frac{\partial}{\partial X_i} \langle U_i'\, C'^2 \rangle - 2 \langle U_i C' \rangle \frac{\partial \langle C \rangle}{\partial X_i} + 2 \langle C' S(C) \rangle + \nu \nabla^2 \langle C'^2 \rangle - 2\nu \left\langle \frac{\partial C'}{\partial X_i} \frac{\partial C'}{\partial X_i} \right\rangle \tag{2.34}$$

Turbulent transport is modeled by gradient diffusion:

$$\langle U_i'\, C' \rangle = - \Gamma_T \frac{\partial \langle C \rangle}{\partial X_i} \tag{2.35}$$

and

$$\langle u_i' C'^2 \rangle = - \Gamma_T \frac{\partial \langle C'^2 \rangle}{\partial X_i} \tag{2.36}$$

and the scalar dissipation is modeled by

$$2 \nu \left\langle \frac{\partial C'}{\partial X_i} \frac{\partial C'}{\partial X_i} \right\rangle = \omega \langle C'^2 \rangle . \tag{2.37}$$

Since the flame is assumed to be statistically one dimensional, derivatives with respect to X_2 and X_3 are zero and the equation can be written in terms of X_1, or X. Therefore, equation (2.34) becomes

$$\frac{\partial \langle C'^2 \rangle}{\partial t} + \langle U \rangle \frac{\partial \langle C'^2 \rangle}{\partial X} = \frac{\partial}{\partial X} \Gamma_T \frac{\partial \langle C'^2 \rangle}{\partial X} + 2 \Gamma_T \left(\frac{\partial \langle C \rangle}{\partial X} \right)^2 + 2 \langle C' S(C) \rangle - \omega \langle C'^2 \rangle \tag{2.38}$$

where ω and Γ_T are defined in Equations (2.3) and (2.4). This model assumes a Taylor expansion of $\langle S(C) \rangle$ up to and including second order terms:

$$\langle S(C) \rangle = S(\langle C \rangle) + \frac{\partial^2 S(\langle C \rangle)}{\partial C^2} \frac{\langle C'^2 \rangle}{2!} \tag{2.39}$$

and

$$\langle C' S(C) \rangle = \frac{\partial S(\langle C \rangle)}{\partial C} \langle C'^2 \rangle . \tag{2.40}$$

Therefore, the transport equations for $\langle C \rangle$ and $\langle C'^2 \rangle$ are

$$\frac{\partial \langle C \rangle}{\partial t} + \langle U \rangle \frac{\partial \langle C \rangle}{\partial X} = \frac{\partial}{\partial X} \Gamma_T \frac{\partial \langle C \rangle}{\partial X} + S(\langle C \rangle) + \frac{\partial^2 S(\langle C \rangle)}{\partial C^2} \frac{\langle C'^2 \rangle}{2!} \tag{2.41}$$

and

$$\frac{\partial \langle C'^2 \rangle}{\partial t} + \langle U \rangle \frac{\partial \langle C'^2 \rangle}{\partial X} = \frac{\partial}{\partial X} \Gamma_T \frac{\partial \langle C'^2 \rangle}{\partial X} + 2\Gamma_T \frac{\partial \langle C \rangle}{\partial X} \frac{\partial \langle C \rangle}{\partial X}$$

$$+ 2 \frac{\partial S(\langle C \rangle)}{\partial C} \langle C'^2 \rangle - \omega \langle C'^2 \rangle \, . \qquad (2.42)$$

These equations can be normalized by ℓ_ϵ and u'. Defining

$$X^* \equiv X/\ell_\epsilon \qquad (2.43)$$

$$U^* \equiv \langle U \rangle / u' \qquad (2.44)$$

$$t^* \equiv (u'/\ell_\epsilon) t \qquad (2.45)$$

$$S^*(C) \equiv (\ell_\epsilon/u') S(C) \qquad (2.46)$$

and using the definitions in Equations (2.5) and (2.6) for Γ_T^* and ω^* the normalized equations for $\langle C \rangle$ and $\langle C'^2 \rangle$ can be written as

$$\frac{\partial \langle C \rangle}{\partial t^*} + \langle U^* \rangle \frac{\partial \langle C \rangle}{\partial X^*} = \frac{C_\mu}{\sigma_\phi} \frac{\partial}{\partial X^*} \frac{\partial \langle C \rangle}{\partial X^*} + S^*(\langle C \rangle)$$

$$+ \frac{\partial^2 S^*(\langle C \rangle)}{\partial C^2} \frac{\langle C'^2 \rangle}{2!} \qquad (2.47)$$

and

$$\frac{\partial \langle C'^2 \rangle}{\partial t^*} + \langle U^* \rangle \frac{\partial \langle C'^2 \rangle}{\partial X^*} = \frac{C_\mu}{\sigma_\phi} \frac{\partial}{\partial X^*} \frac{\partial \langle C'^2 \rangle}{\partial X^*} + \frac{C_\mu}{\sigma_\phi} \frac{\partial \langle C \rangle}{\partial X^*} \frac{\partial \langle C \rangle}{\partial X^*}$$

$$+ 2 \frac{\partial S^*(\langle C \rangle)}{\partial C} \langle C'^2 \rangle - C_\phi \langle C'^2 \rangle \qquad (2.48)$$

where $\langle U^* \rangle$ is the mean flame speed.

These equations are solved for $\langle C \rangle$ and $\langle C'^2 \rangle$ using the finite difference technique described in Appendix A. The finite difference grid is similar to the grid described for the Monte Carlo model. Each node has a value of $\langle C \rangle$ and $\langle C'^2 \rangle$ associated with it and the flame thickness is defined by Equation (1.8).

CHAPTER 3

In this chapter, calculations based on the two models described in Chapter 2 are reported. The calculations are for a Reynolds number of 10^6 and for a range of Damkohler numbers. The results reported include turbulent flame speeds, thicknesses and profiles of $\langle C \rangle$ and $\langle C'^2 \rangle$.

From Damkohler's theory, it has been shown in Chapter 1 (see Equations (1.29) and (1.30)) that the turbulent flame speed and thickness are

$$S_T^* = \frac{S_T}{u'} = (\Omega\,(C_\mu + 1/R))^{1/2}\, S_\ell^* \qquad (3.1)$$

and

$$(\ell_T^*) = \frac{\ell_T}{\ell_\epsilon} = ((C_\mu + 1/R)/\Omega)^{1/2}\, \ell_\ell^* . \qquad (3.2)$$

These theoretical values of flame speed and thickness, denoted by $(S_T^*)_D$ and $(\ell_T^*)_D$, are compared to the calculations from the Monte Carlo and moment models.

The moment model reduces to the Damkohler theory as Ω approaches zero. Therefore, values of flame speed, $(S_T^*)_M$, and flame thickness, $(\ell_T^*)_M$, calculated from the moment model approach $(S_T^*)_D$ and $(\ell_T^*)_D$ as the Damkohler number decreases. For small Damkohler numbers, calculated values of $(S_T^*)_M$ and $(\ell\ )_M$ can

be used to determine S_ℓ^* and ℓ_ℓ^* from Equations (3.1) and (3.2). Figures 9 and 10 show plots of flame speed and flame thickness calculated from the moment model. For Damkohler numbers of 0.01 and less, S_ℓ^* and ℓ_ℓ^* are determined to be 0.768 and 4.78, respectively. As in previous chapters, these values of S_ℓ^* and ℓ_ℓ^* are used in all subsequent calculations. Figure 11 shows the ratios of $(S_T^*)_M/(S_T^*)_D$ and $(\ell_T^*)_M/(\ell_T^*)_D$ as a function of Ω. These ratios are calculated using Equations (3.1) and (3.2) and the results from Figures 9 and 10. The moment model departs from the Damkohler theory for Damkohler numbers greater than 0.1 and the model fails to reach the convergence criterion described in Appendix A for values of Ω greater than 0.165.

The moment model assumes small fluctuations of C about the mean (i.e. $\langle C'^2 \rangle \ll 1.0$). Figure 12 shows a plot of the maximum value of $\langle C'^2 \rangle^{1/2}$, or C'_{max}, versus Ω. As Ω increases, C'_{max} increases. Since C' is nonzero and the reaction rate (see Equation (1.13)) is highly nonlinear, the assumption of small fluctuations of C about the mean is not valid for large Damkohler numbers. Figure 13 shows contours of the reaction rate $S^*(\langle C \rangle)$ and

$$\frac{\partial^2 S^*(\langle C \rangle)}{\partial C^2} \quad \frac{\langle C'^2 \rangle}{2!}$$

versus $\langle C \rangle$ for $\Omega = 0.165$. The second derivative term is much larger than $S^*(\langle C \rangle)$ for small values of $\langle C \rangle$. This region of

the flame is characterized by small changes in <C> and is the preheat zone. Chemical reaction is limited by the growth of this zone.

Figures 14 and 15 show the results of calculations from the Monte Carlo model of flame speed, $(S_T^*)_{MC}$, and flame thickness, $(\ell_T^*)_{MC}$, as a function of the Damkohler number. The figures shows the mean values of $(S_T^*)_{MC}$ and $(\ell_T^*)_{MC}$ (indicated by dark squares) with plus and minus one standard deviation (indicated by the circles and triangles). The standard deviation σ of $(S_T^*)_{MC}$ is calculated from the values of S_T obtained in M independent trials,

$$\sigma = \left(\frac{1}{M} \sum_{i=1}^{M} \left[(S_T^*)_{MC} \right]^2 - \langle (S_T^*)_{MC} \rangle^2 \right)^{1/2} \qquad (3.3)$$

where $((S_T^*)_{MC})_i$ is the value of the ith trial. For a given value of Ω, the flame speed and flame thickness are calculated M times so that the standard error is

$$\text{error} = \frac{\sigma}{\sqrt{M}} \quad . \qquad (3.4)$$

For the calculations shown in Figures 14 and 15, N is 25. Therefore, the likely error of $(S_T^*)_{MC}$ is one fifth of the standard deviation.

For simplicity, these results are redrawn in Figure 16 and 17 using only the mean values. Figure 16 also shows the

curves of flame speed and flame thickness obtained from the Damkohler theory and laminar flame theory (Equations (1.11) and (1.12)). Figure 17 shows values of ℓ_T^* from the Damkohler theory and calculated values of $(\ell_T^*)_{MC}$. For a Reynolds number of 10^6, the laminar flame thickness is of the order one millionth and is not shown in Figure 17. The Monte Carlo model departs from the Damkohler theory for values of the Damkohler number greater than 0.1. Below Ω=0.1, $(S_T^*)_{MC}$ and $(\ell_T^*)_{MC}$ are within one standard deviation of $(S_T^*)_D$ and $(\ell_T^*)_D$. Both $(S_T^*)_{MC}$ and $(\ell_T^*)_{MC}$ depart from the Damkohler theory and approach horizontal asymptotes as Ω increases. These asymptotes are characteristic of Regions I and II of Figure 6 which is described in Chapter 1. According to Figure 8, $(S_T^*)_{MC}$ should depart from the Damkohler theory and tend to a horizontal asymptote as Ω increases. As may be seen in Figure 16, $(S_T^*)_{MC}$ does reach such an asymptote but not until $\Omega=10^4$. This result can be explained by examining the pdf's of C for a range of Damkohler numbers.

Figure 18 shows the calculated pdf's of C at five different grid locations for a Damkohler number of 10.0. The mean concentration of C at each grid location is indicated by an arrow. For $\langle C\rangle$=0.5, the magnitude of a spike at ψ=1.0 represents the probability of fully burnt fluid at $X_{0.5}^*$. As $\langle C\rangle$ increases, the probability of fluid with a concentration of 1.0

increases. Figure 19 shows the pdf's at $X^*_{0.5}$ for five different Damkohler numbers. In the limit of zero Damkohler number, there are no fluctuations and so the pdf is a Dirac delta function at <C>. For the smallest Damkohler number shown, Ω =0.05, the pdf is a bimodal distribution. As Ω approaches zero, the shape of the pdf will approach a delta function at <C>=0.5. As Ω increases the pdf changes from the bimodal distribution to two spikes located at ψ=0.0, and ψ=1.0. This change is due to the increase in the chemical reaction rate. As the reaction rate increases, the probability of fully burnt fluid increases. Also, the probability of intermediate values of C (in the range $0.0< \psi <1.0$) decreases. Since a fluid with a concentration of 0.0 cannot react (i.e. $S^*(0.0)= 0.0$), reaction does not occur until after the mixing process begins. As suggested by Figure 19, once fluid reaches a certain value of C, say C^*, complete reaction occurs almost instantaneously. Therefore, the probability of fluid with a concentration in the range $C^*< \psi <1.0$ is very small. This value of C^* is dependent on Ω and, as Ω increases, C^* decreases. For larger values of Ω , although the chemical reaction is very fast, the Monte Carlo model is mixing limited since reaction does not occur until after mixing.

The Monte Carlo model assumes the steepest scalar gradients are due to scalar dissipation caused by turbulent

straining. This assumption is invalid when laminar flamlets give rise to even steeper scalar gradients. Therefore, the Monte Carlo model does not apply in the laminar flame region of the R-Ω plane and the calculated flame speeds and flame thicknesses do not match S_T^* and ℓ_T^* from laminar flame theory.

Figures 20,21, and 22 compare the results of the moment model to the Monte Carlo model. Although it would have been desirable to have more data points in Figures 20 and 21, conclusions can be drawn from the data shown. In Figure 20, the ratios of $(S_T^*)_{MC}/(S_T^*)_D$ and $(S_T^*)_M/(S_T^*)_D$ are plotted in terms of Ω . Both models depart from the Damkohler theory for Damkohler numbers greater than 0.1, but the moment model predicts a greater value of S_T^* than the Damkohler theory while the Monte Carlo model predicts a smaller value of S_T. In Figure 21, the ratios of $(\ell_T^*)_{MC}/(\ell_T^*)_D$ and $(\ell_T^*)_M/(\ell_T^*)_D$ are plotted in terms of Ω . For Damkohler numbers greater than 0.1, the Monte Carlo model predicts a larger flame thickness than the Damkohler theory and the moment model predicts a smaller flame thickness.

Figure 22 shows maximum values of C' versus Ω for the Monte Carlo model and the moment model. For Damkohler numbers larger than 0.1, value of C'_{max} calculated from the moment model are larger than the values of C'_{max} calculated from the Monte

Carlo model. Due to these larger values of C'_{max}, the moment model predicts larger values of S_T^* than the Monte Carlo model. The value of C'_{max} according to the Monte Carlo method approaches an asymptotic value of 0.5 as Ω tends to infinity. This is the maximum possible value, corresponding to a double delta function distribution.

The results of the moment model need more explanation. In the limit of zero Damkohler number, the moment model assumes a pdf of a Dirac delta function at <C>. Even for a Damkohler number of 0.05, the pdf calculated from the Monte Carlo model (see Figure 19) is a bimodal distribution. Therefore, this assumption is incorrect and the moment model is invalid.

CONCLUSIONS

Through dimensional analysis, two indepenent, dimensionless parameters are defined. These parameters are the Reynolds number, R, and the Damkohler number, Ω . Any point on the R-Ω plane correpsonds to a set of operating conditions for a turbulent flame. Three different regimes of combustion on the R-Ω plane are defined: the Damkohler theory regime where Damkohler's theory is applicable, the regime where the turbulent flame speed and thickness are independent of R and Ω , and the laminar flame regime.

Two different models are used to predict turbulent flame speeds and thicknesses for a range of Damkohler numbers. Both models are independent of Reynolds number. The first model, the moment model, solves the scalar transport equation and assumes small fluctuations of C about the mean. The moment model fails because it assumes a pdf of a Dirac delta function at <C>. As explained in Chapter 3, this assumption is invalid.

The second model uses a Monte Carlo method to solve the pdf equation of C. It assumes the steepest scalar gradients are due to scalar dissipation caused by turbulent straining. For small values of Ω , the results of the Monte Carlo model match those of the Damkohler theory. For Damkohler numbers

greater than 0.1, the Monte Carlo model departs from the Damkohler theory and reaches an asymptote where the flame speed and flame thickness are independent of R and Ω. For the laminar flame regime, the results of the Monte Carlo model do not match the values of flame speed and flame thickness obtained from laminar flame theory. This is because laminar flamelets cause steeper scalar gradients than assumed by gradient diffusion. From the pdf's calculated by the Monte Carlo model, it is shown that the model is mixing limited.

REFERENCES

(1) Pope, S.B.,"Probability Distributions of Scalars in Turbulent Shear Flow", Second International Symposium on Turbulent Shear Flows, Imperial College London, (1979)

(2) Damkohler, G., Z.Elekrochemie Angewandte, Phys. Chem. 46, 601 (1940) (English transation, NACA TM 1112 (1947).)

(3) Launder, B.E. and Spaulding, D.B., Mathematical Model of Turbulence, Academic Press, 1972.

(4) Pope, S.B., Phil. Trans. R. Soc. Lond. A. 291, 529 (1979).

(5) Curl, R.L., A.I.Ch.E.J. 9, 175 (1963)

(6) Pope, S.B.,"A Monte Carlo Method for the PDF Equations of Turbulent Reactive Flow",Combustion Science and Technology, 25, 159 (1981)

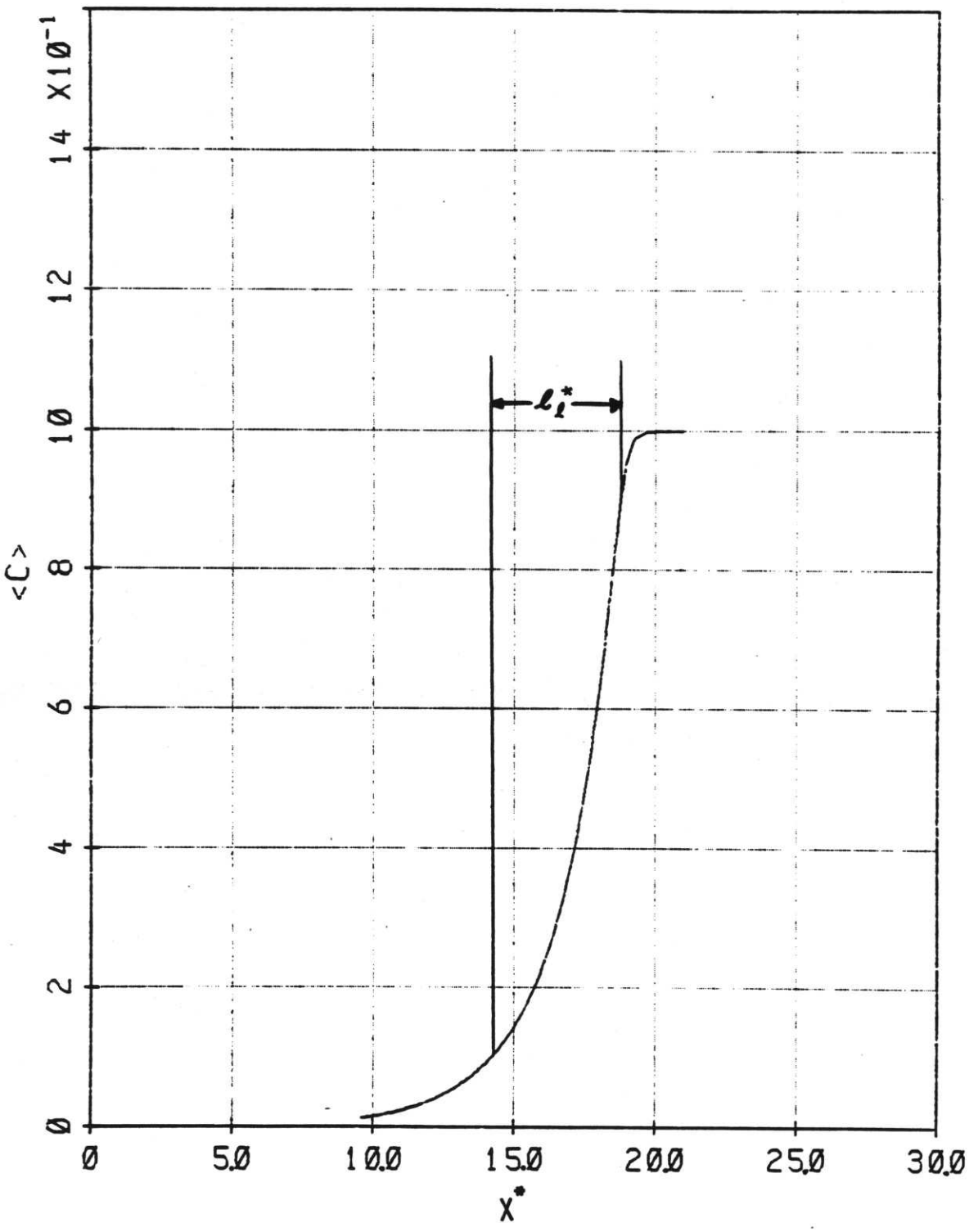

Figure 1 - Graphical Solution of simplified scalar transport equation

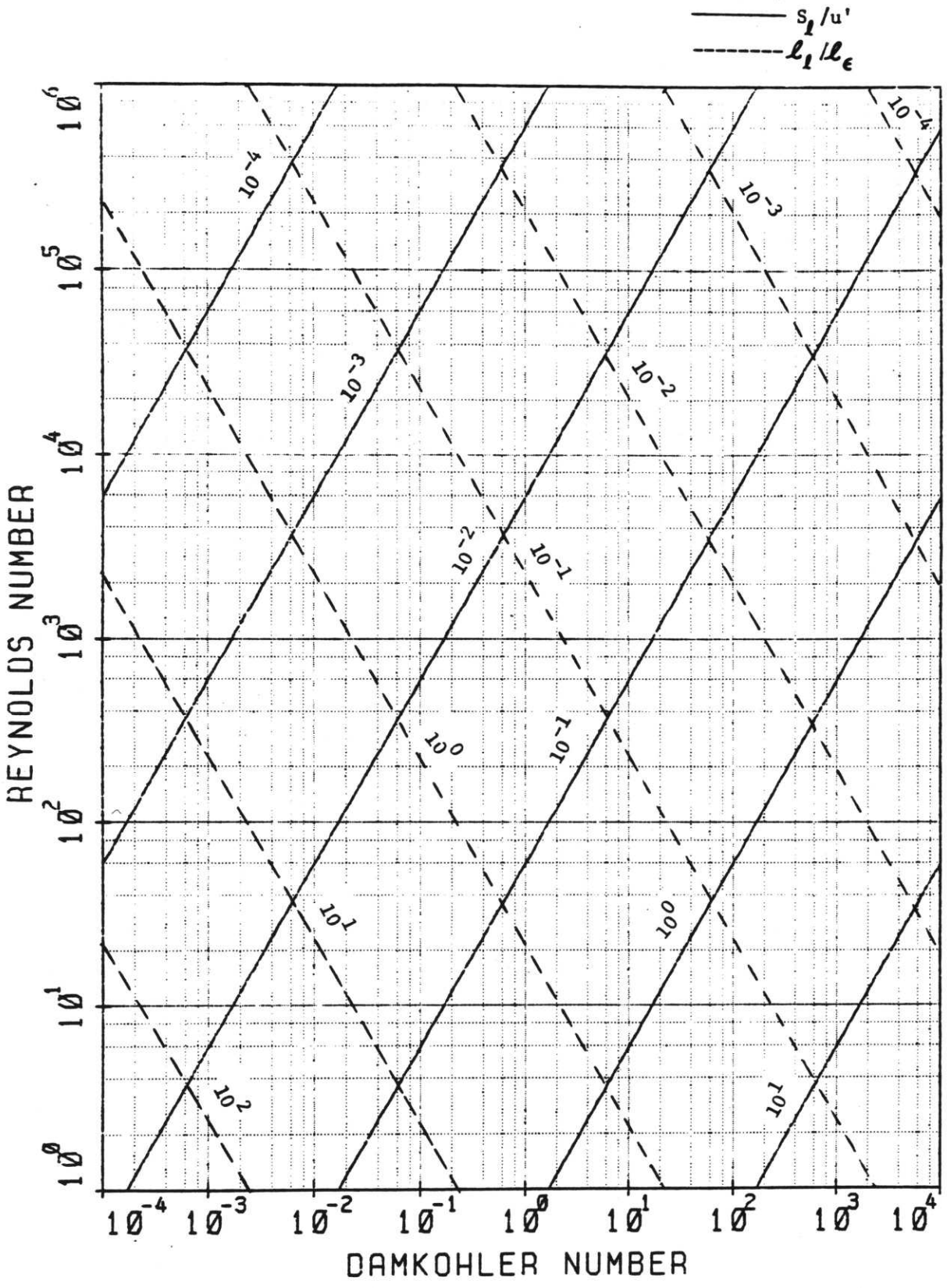

Figure 2- Contours of S_ℓ /u' and ℓ_1/ℓ_ϵ

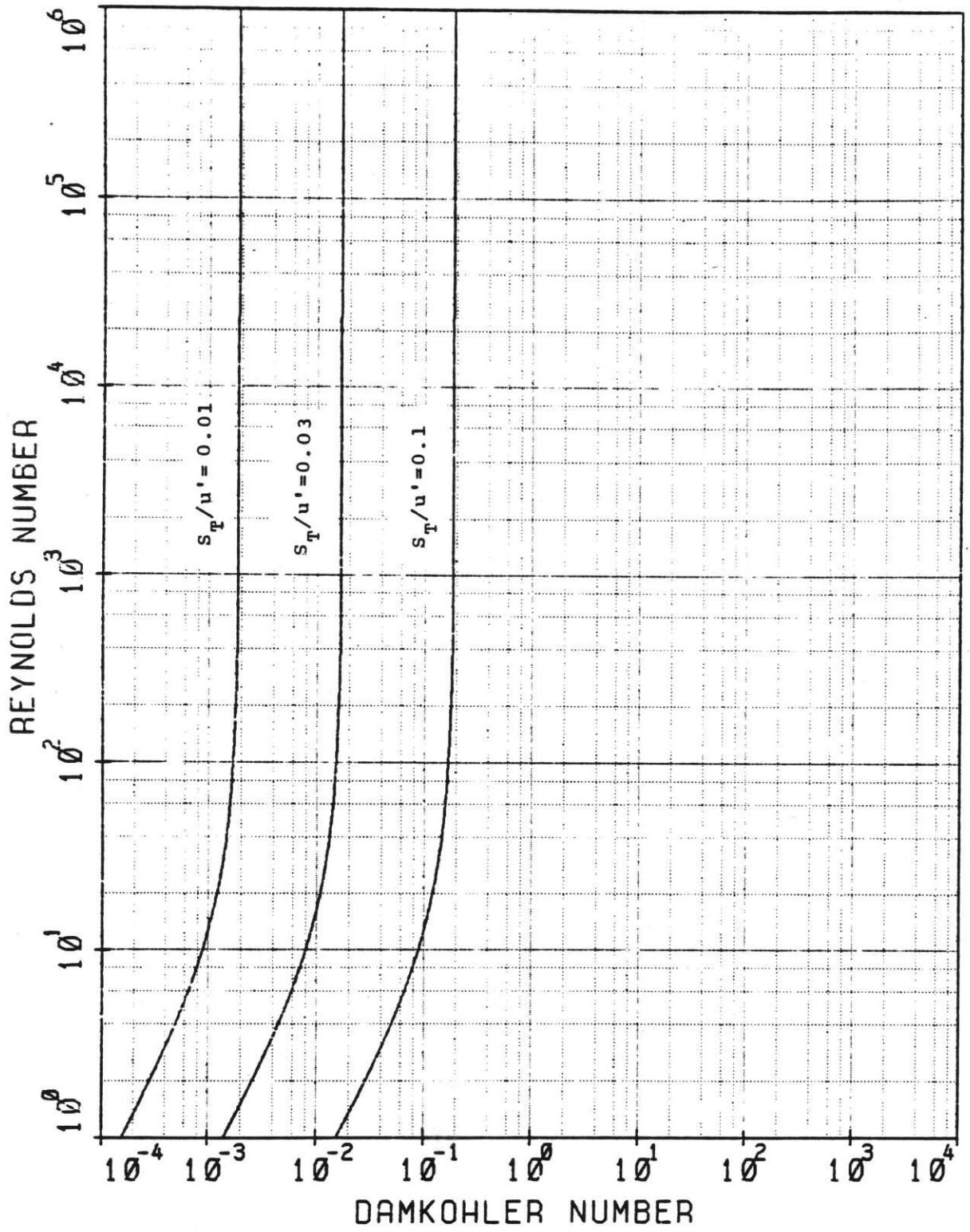

Figure 3- Contours of S_T/u' from Damkohler's theory

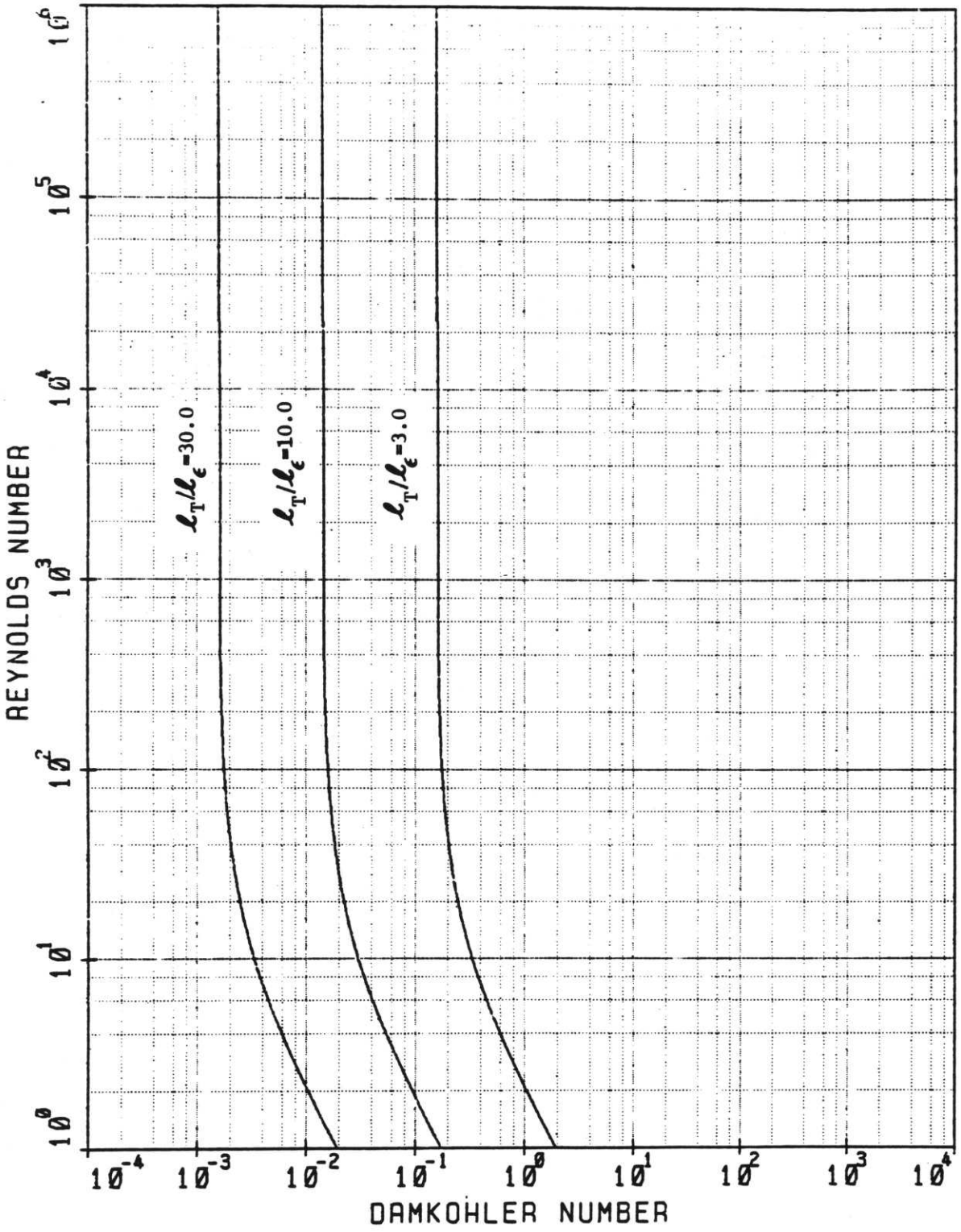

Figure 4 - Contours of ℓ_T/ℓ_ϵ
From Damkohler's Theory

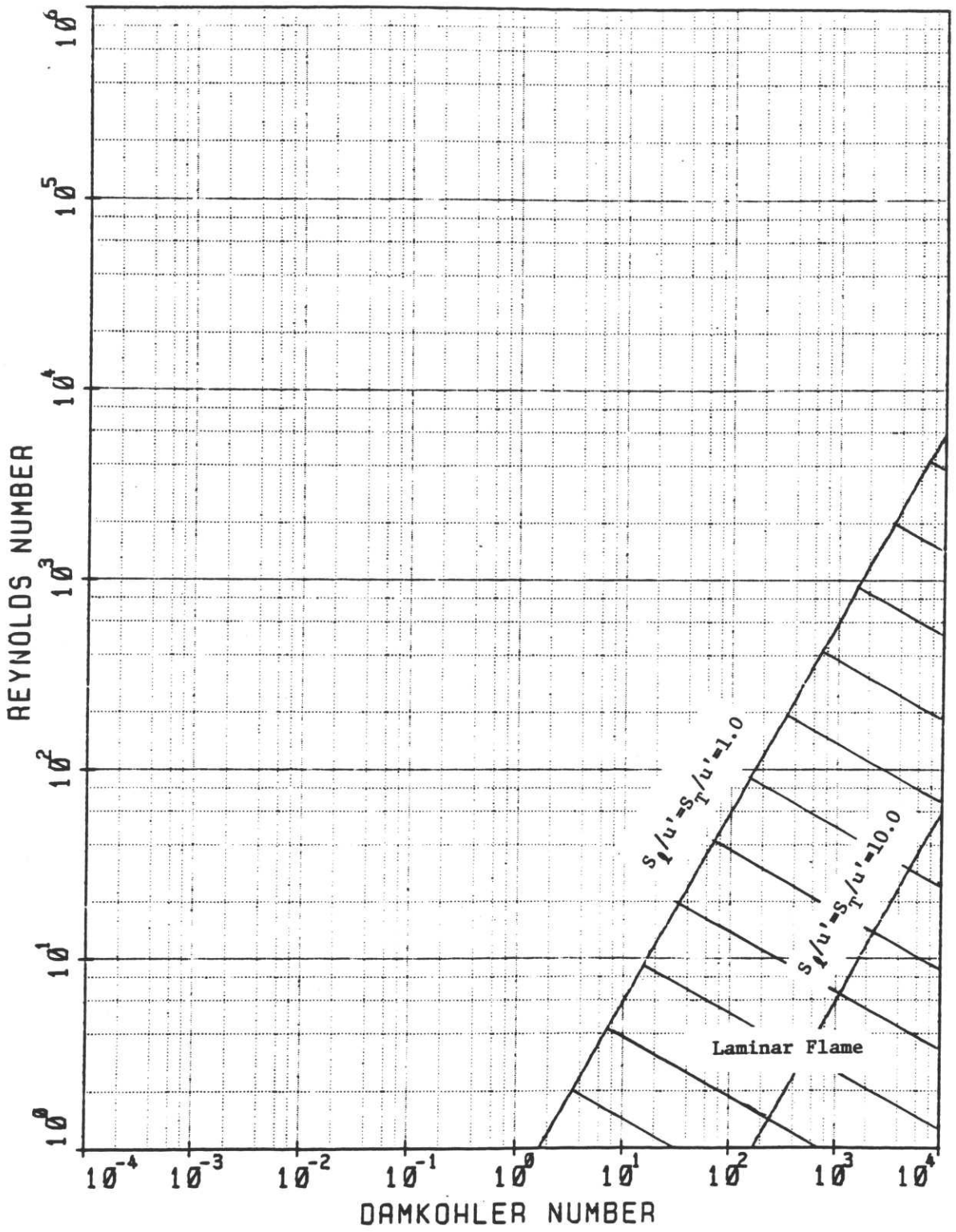

Figure 5 - Laminar Flame Regime

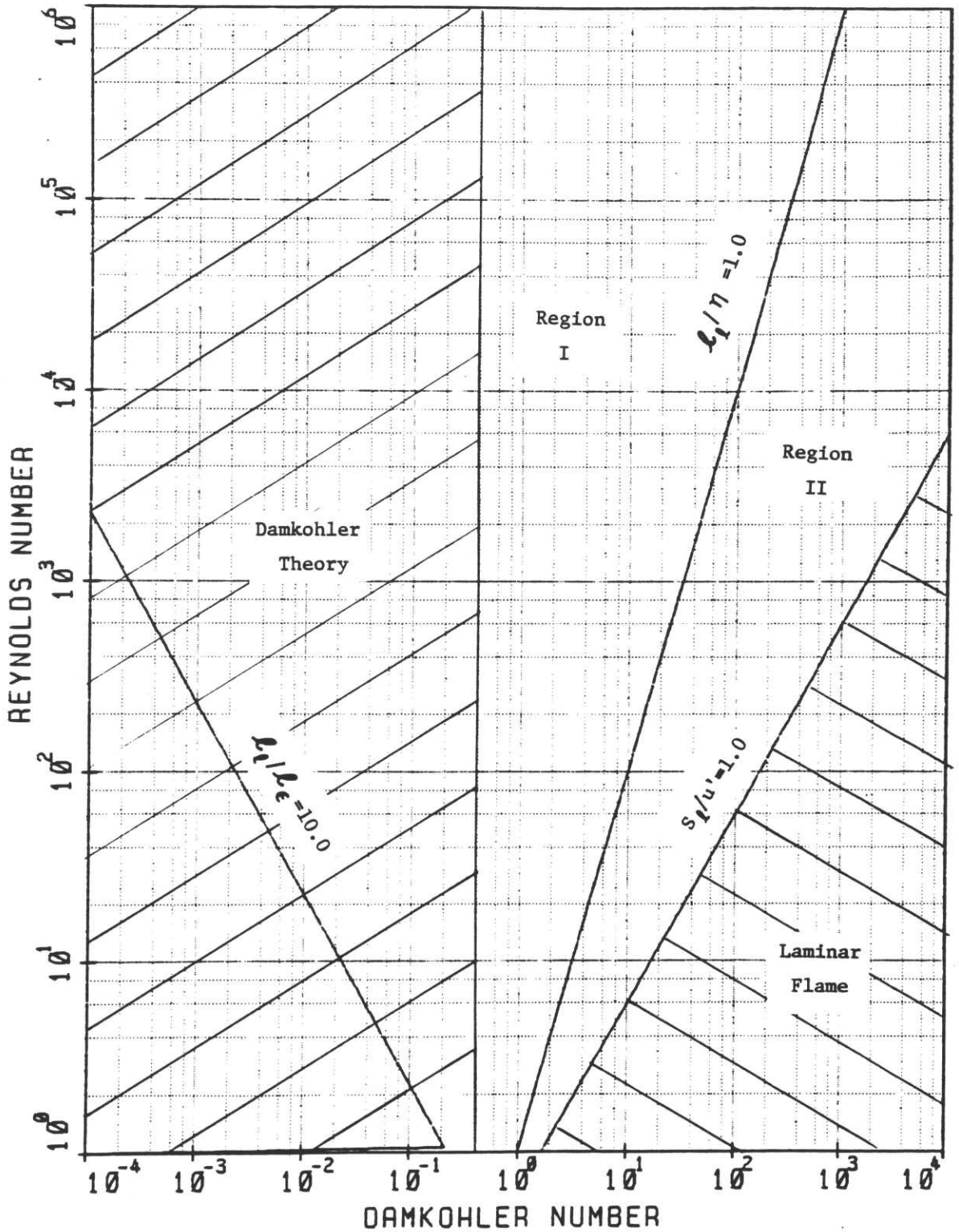

Figure 6 - Regimes of Turbulence on R-Ω plane

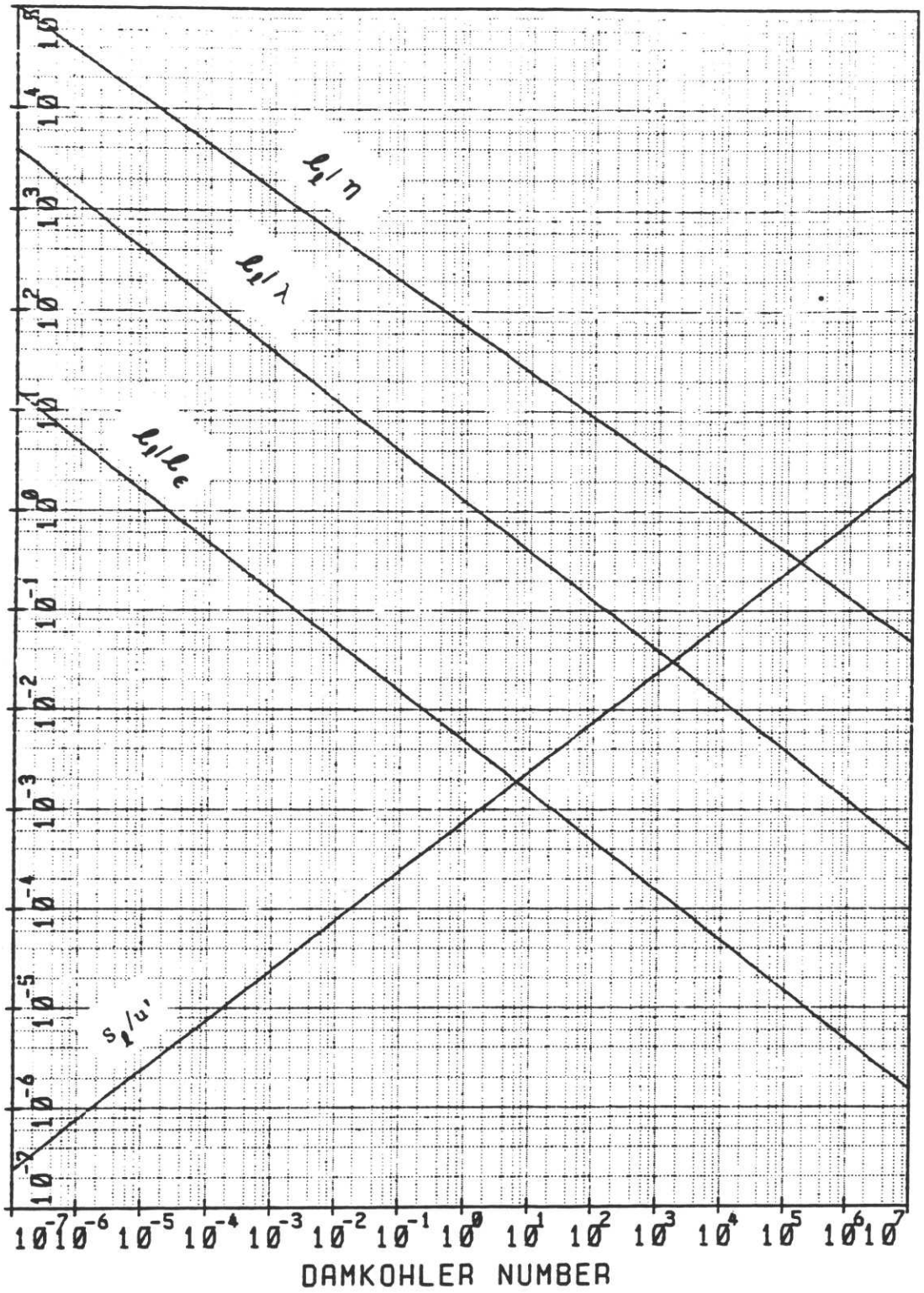

Figure 7 - Several dimensionless parameters plotted against Ω for Reynolds number of 10^6

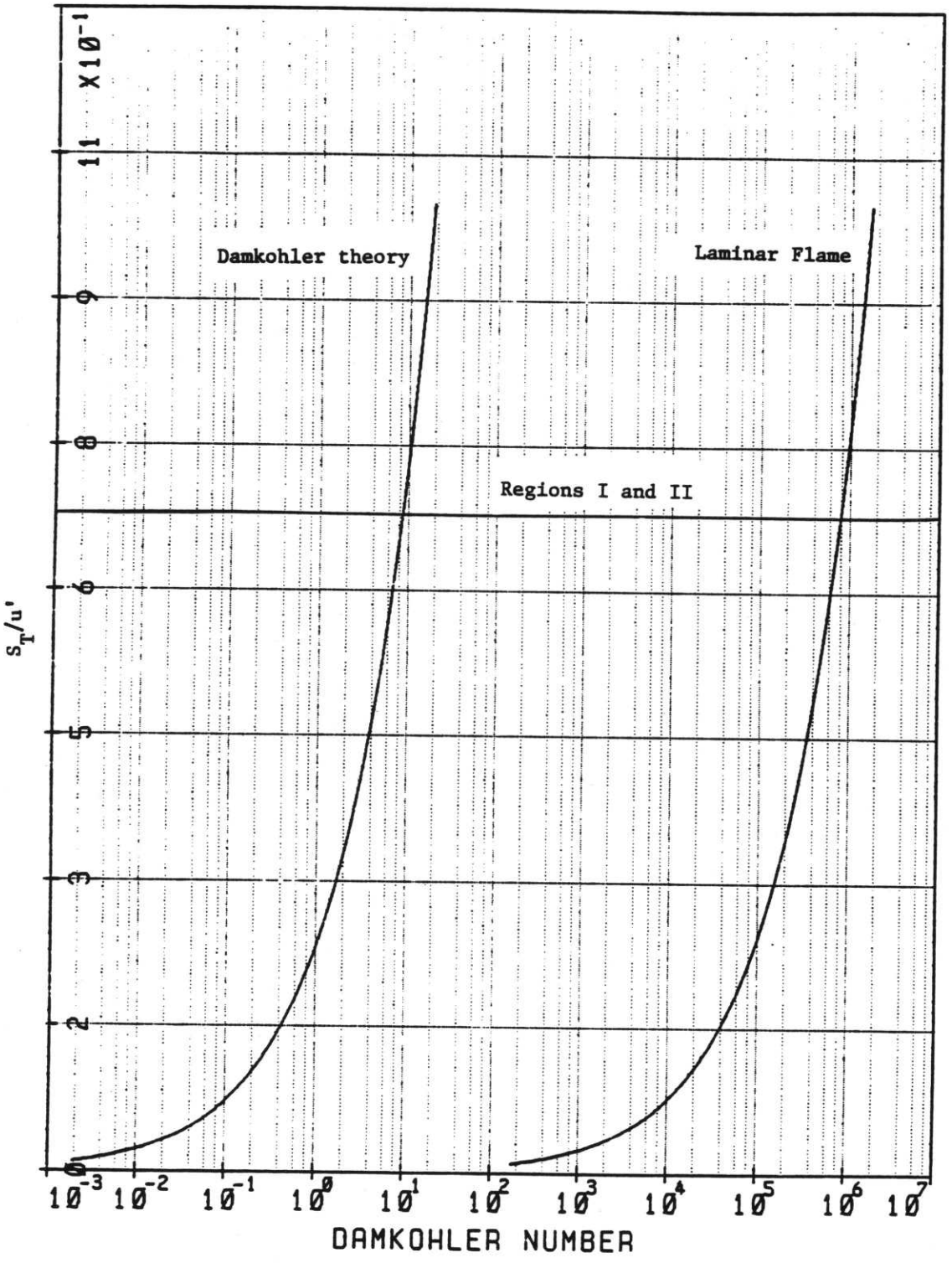

Figure 8 - S_T/u' for Reynolds number of 10^6 for the different regimes of combustion

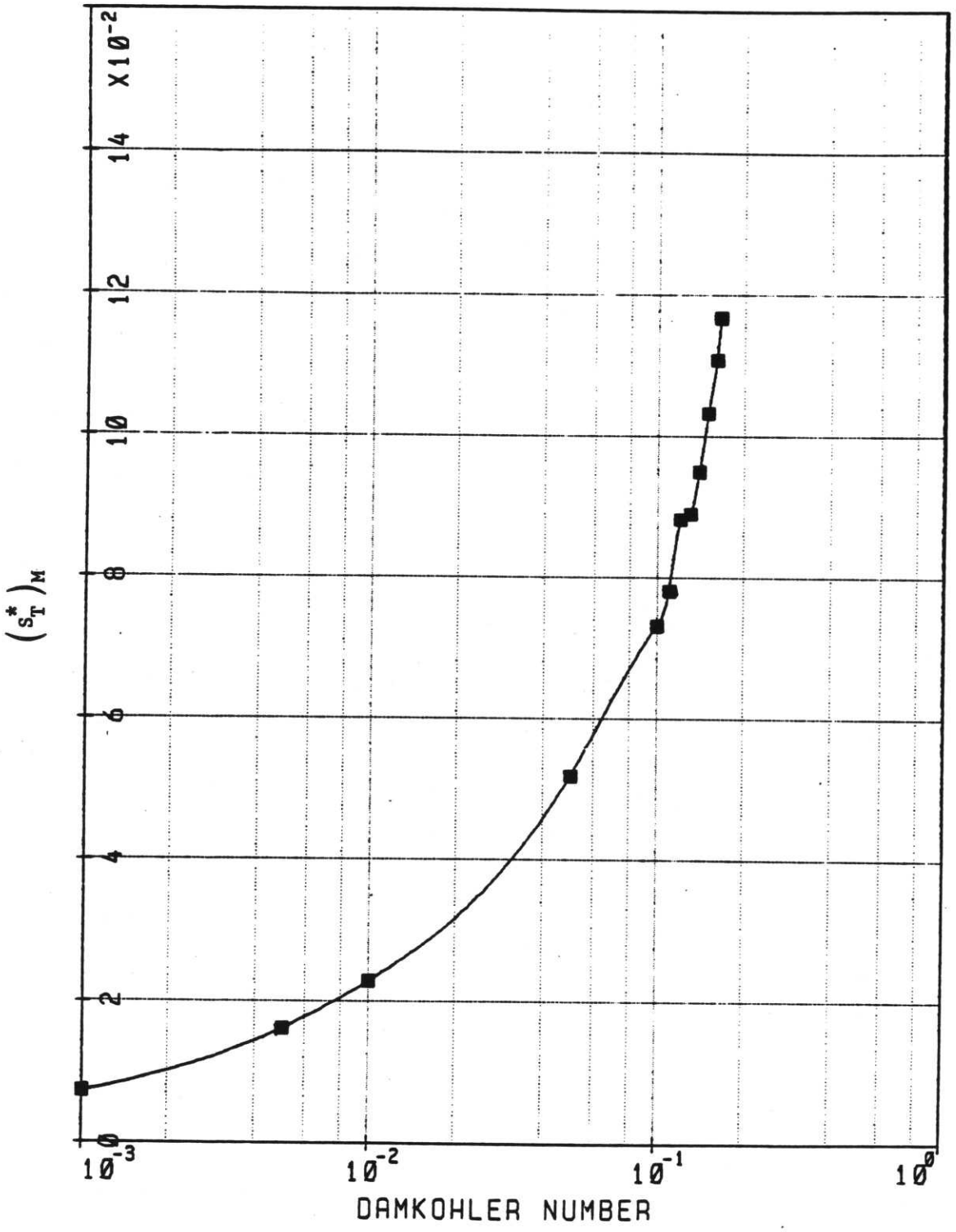

Figure 9 - Turbulent flame speed versus Damkohler number from moment model

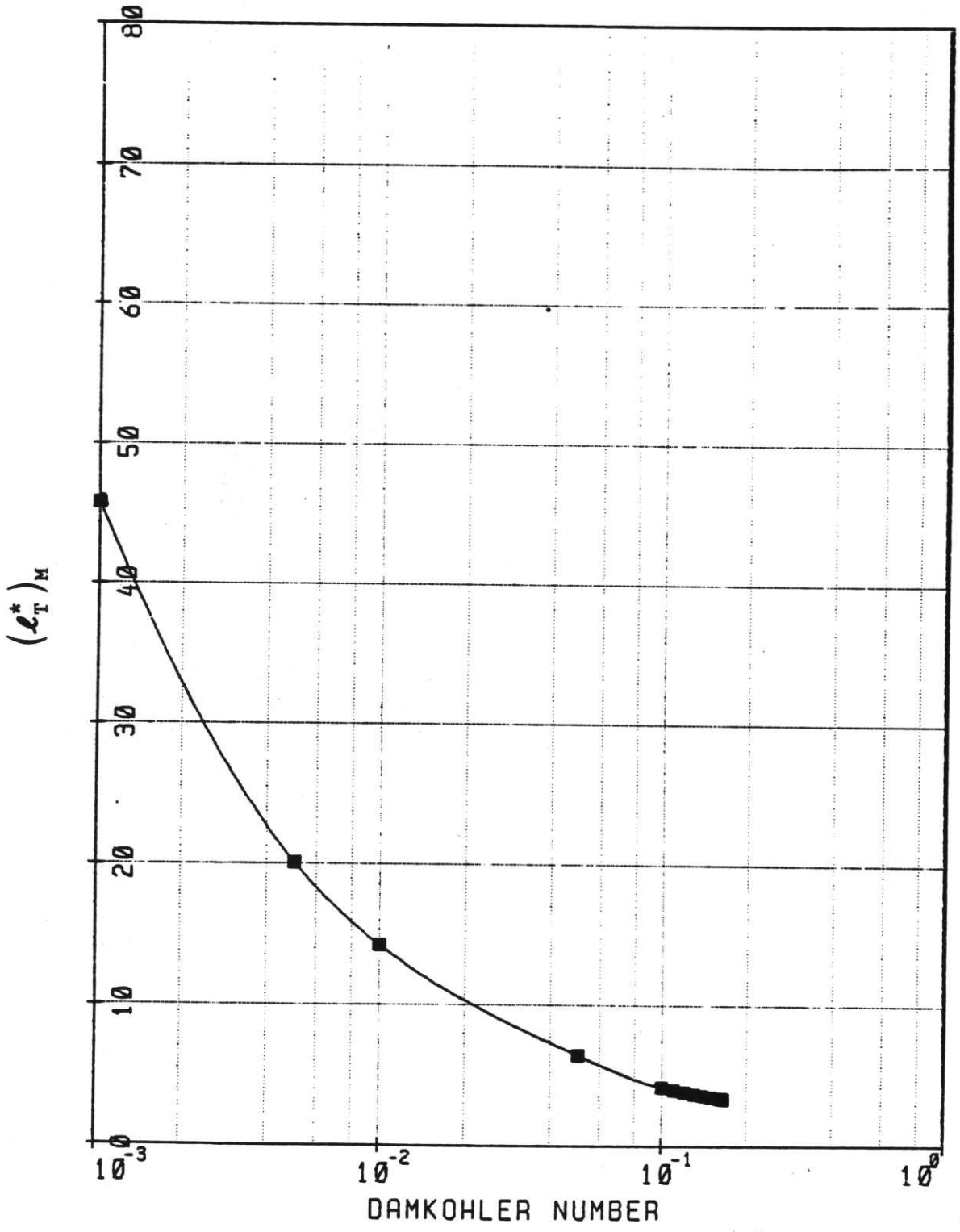

Figure 10 - Turbulent flame thickness versus Damkohler number from moment model

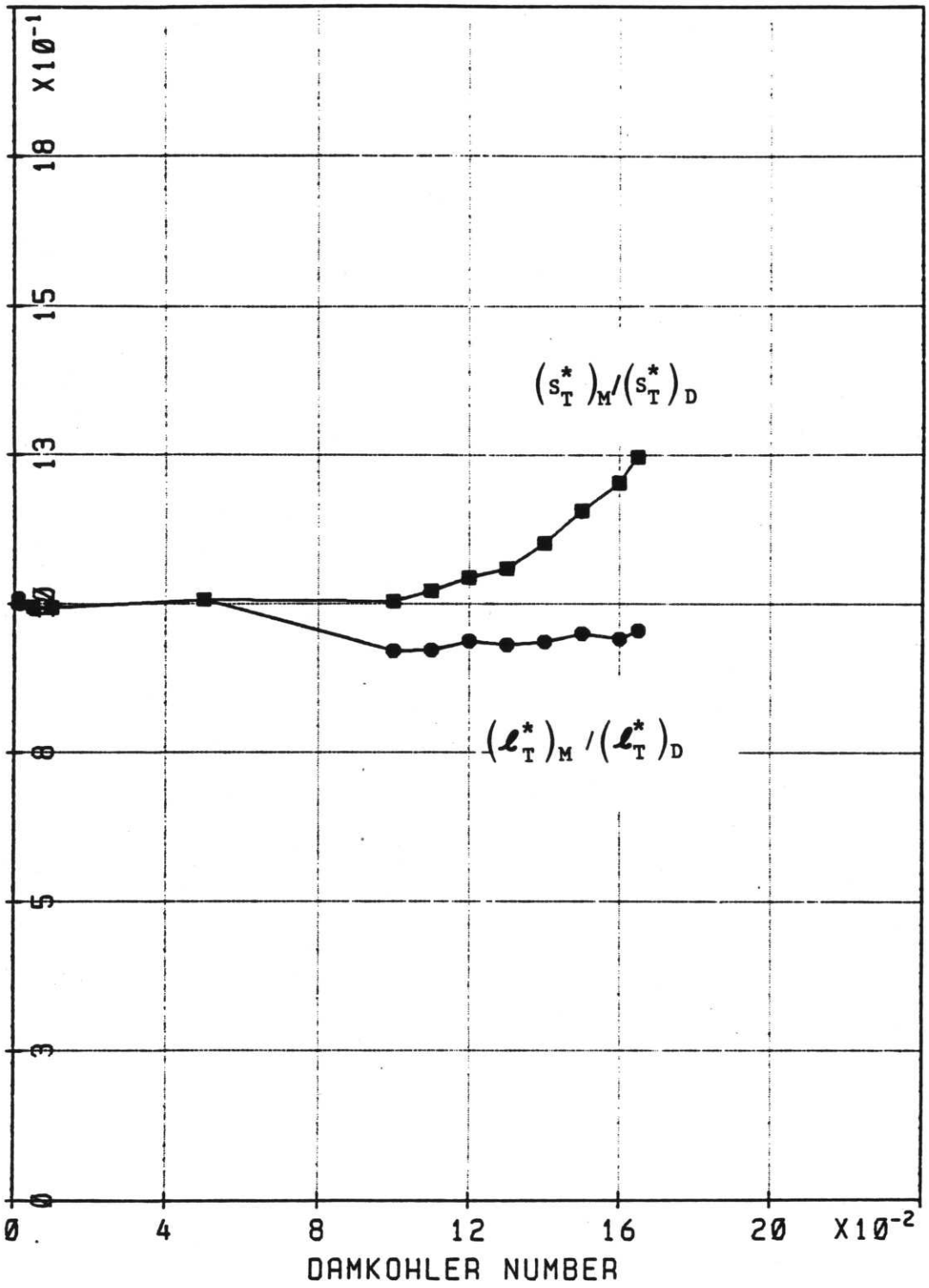

Figure 11 - Ratios of turbulent flame speed and turbulent flame thickness from moment model to turbulent flame speed and thickness from Damkohler's theory

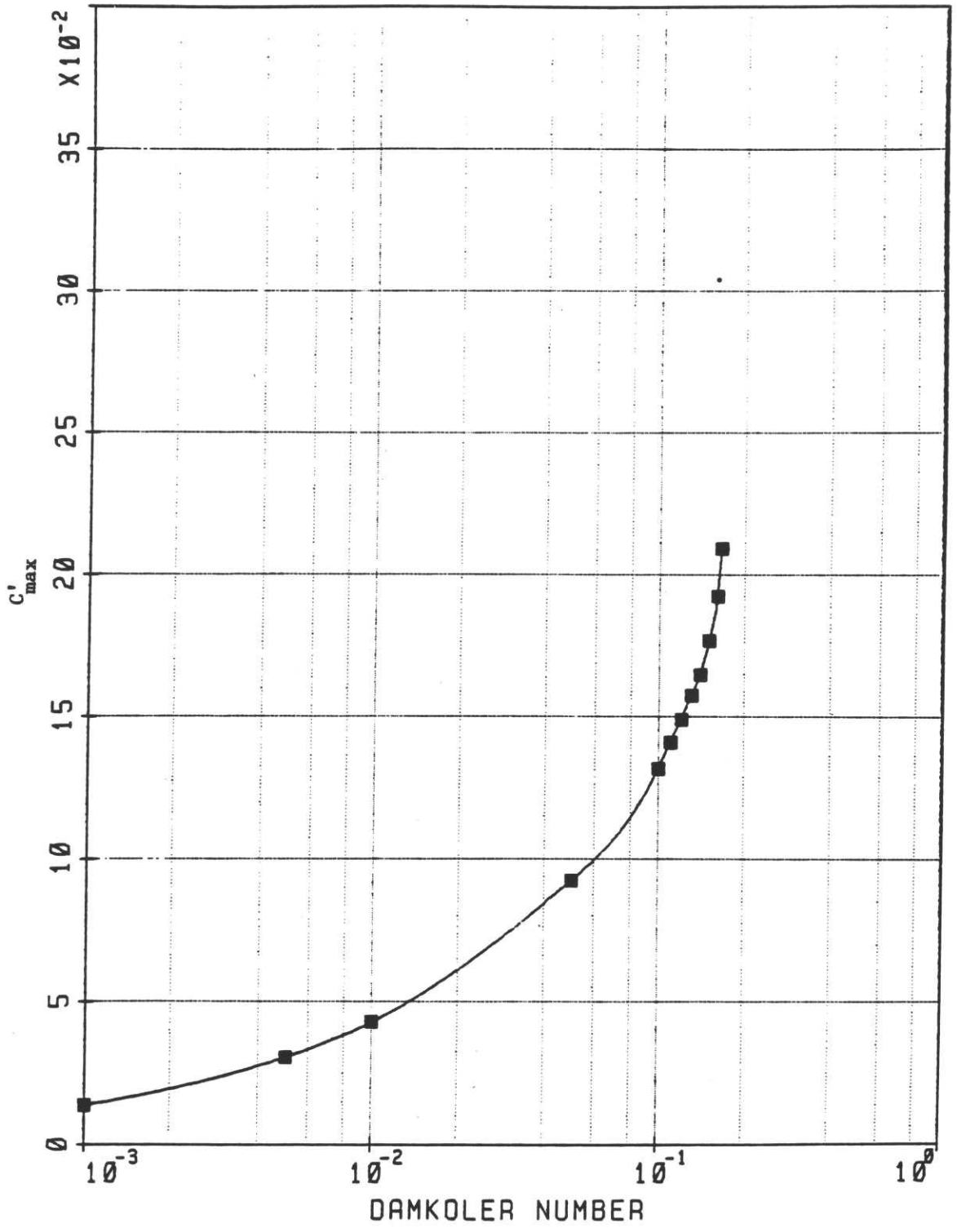

Figure 12 - Values of C'_{max} versus Damkohler number from moment model

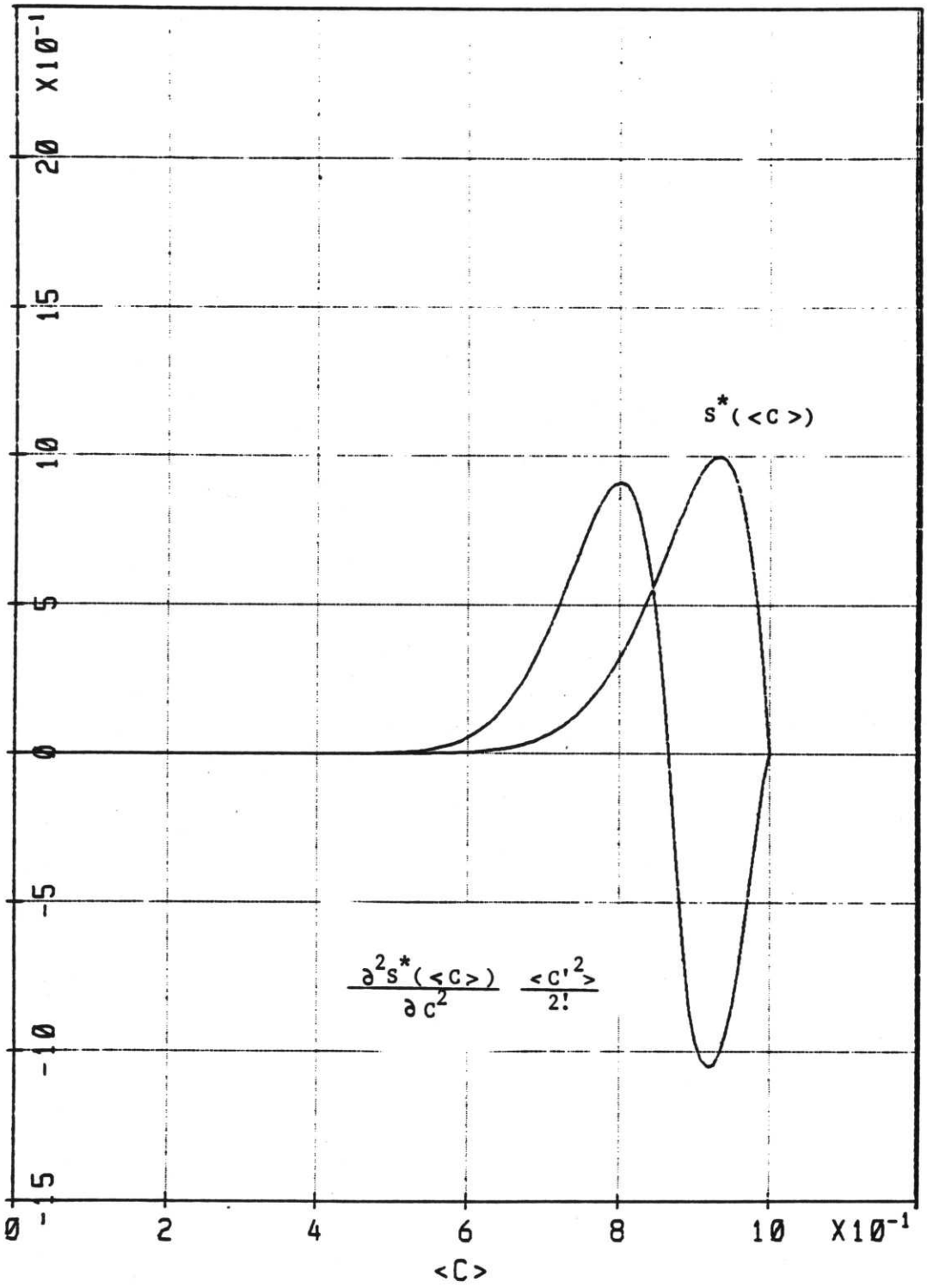

Figure 13 - Contribution of reaction for Damkohler Number of 0.165 from moment model

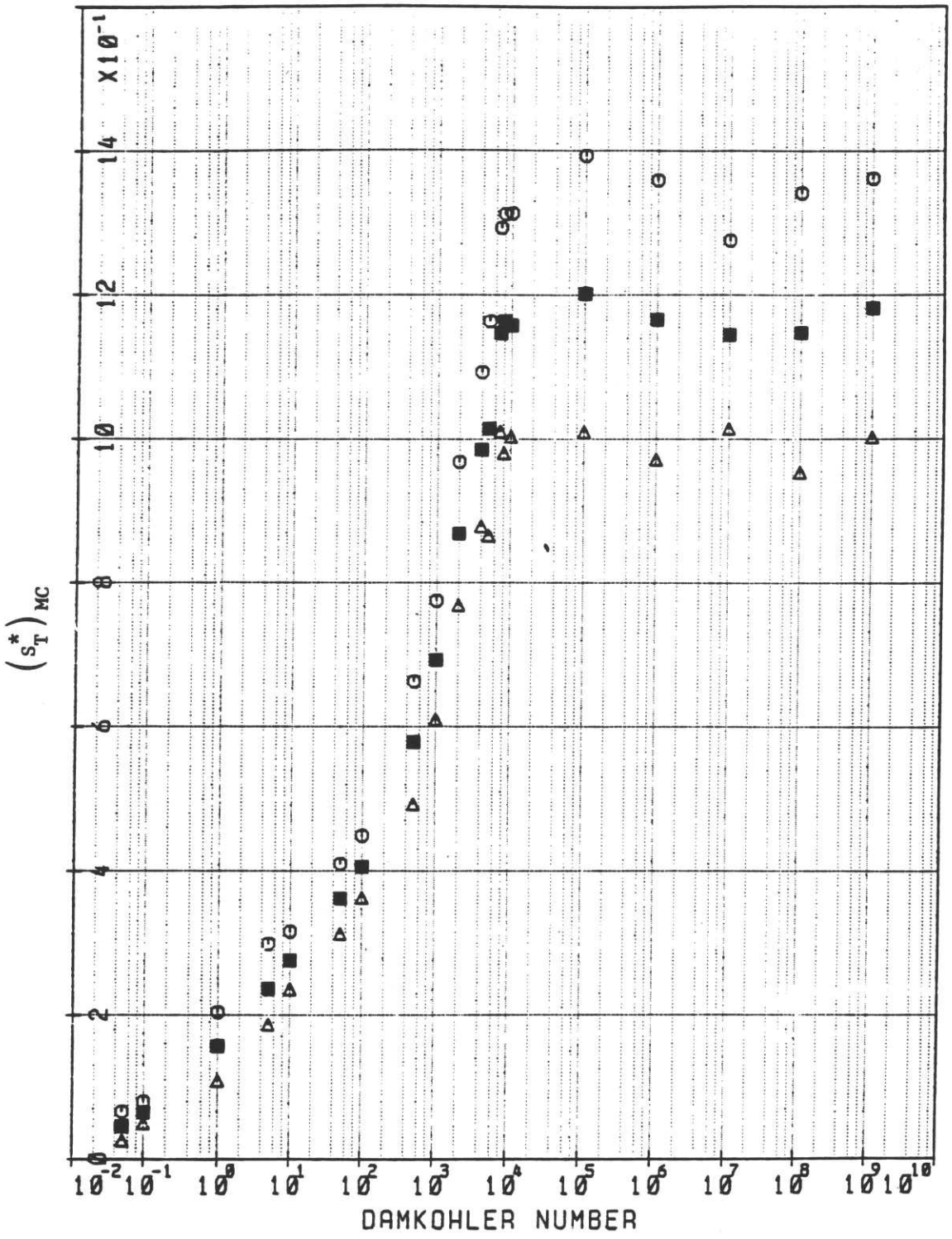

Figure 14- Turbulent flame speed results from Monte Carlo model (mean values are indicated by dark points)

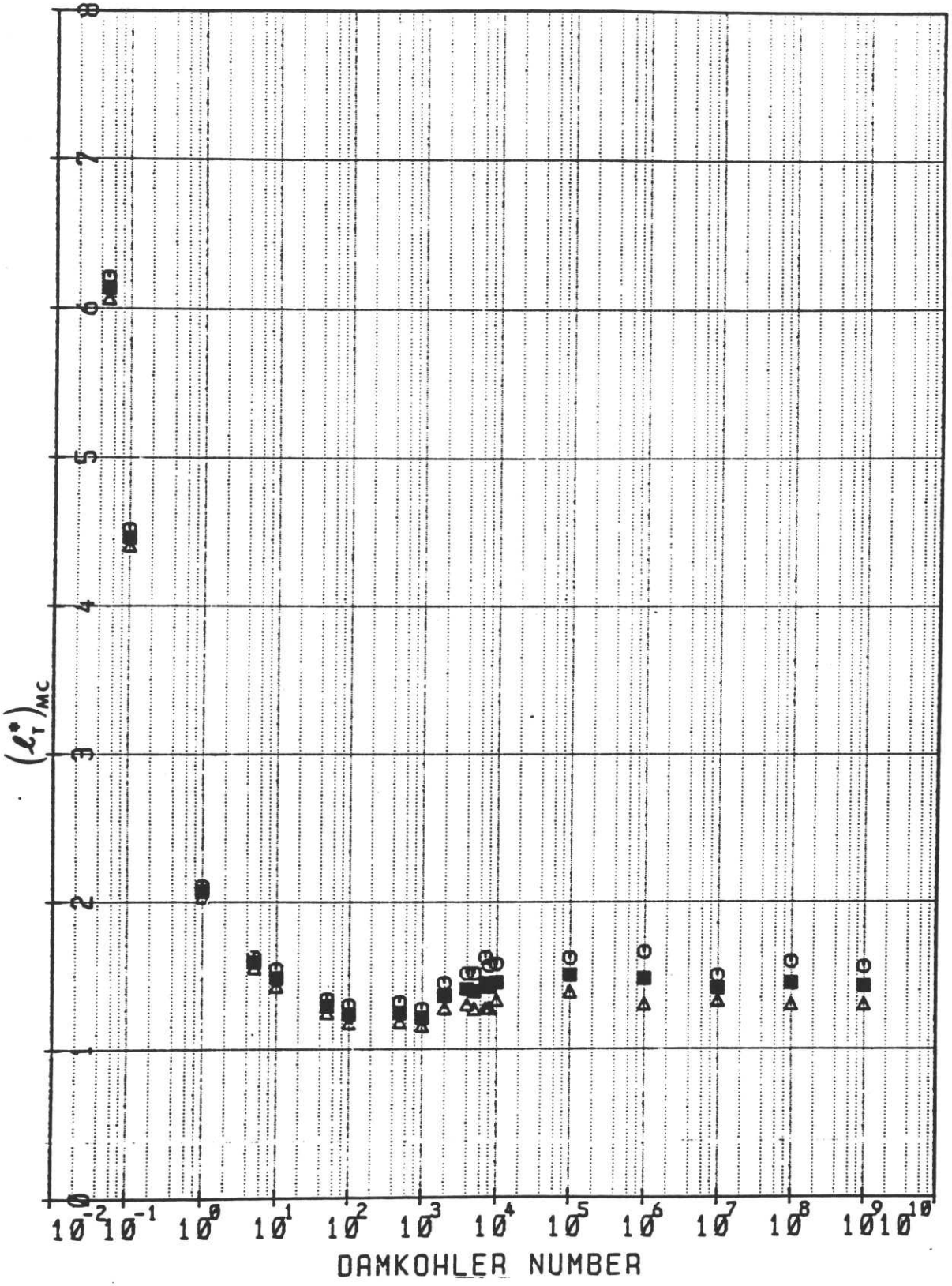

Figure 15 - Calculated values of (ℓ_T^*) from the Monte Carlo model

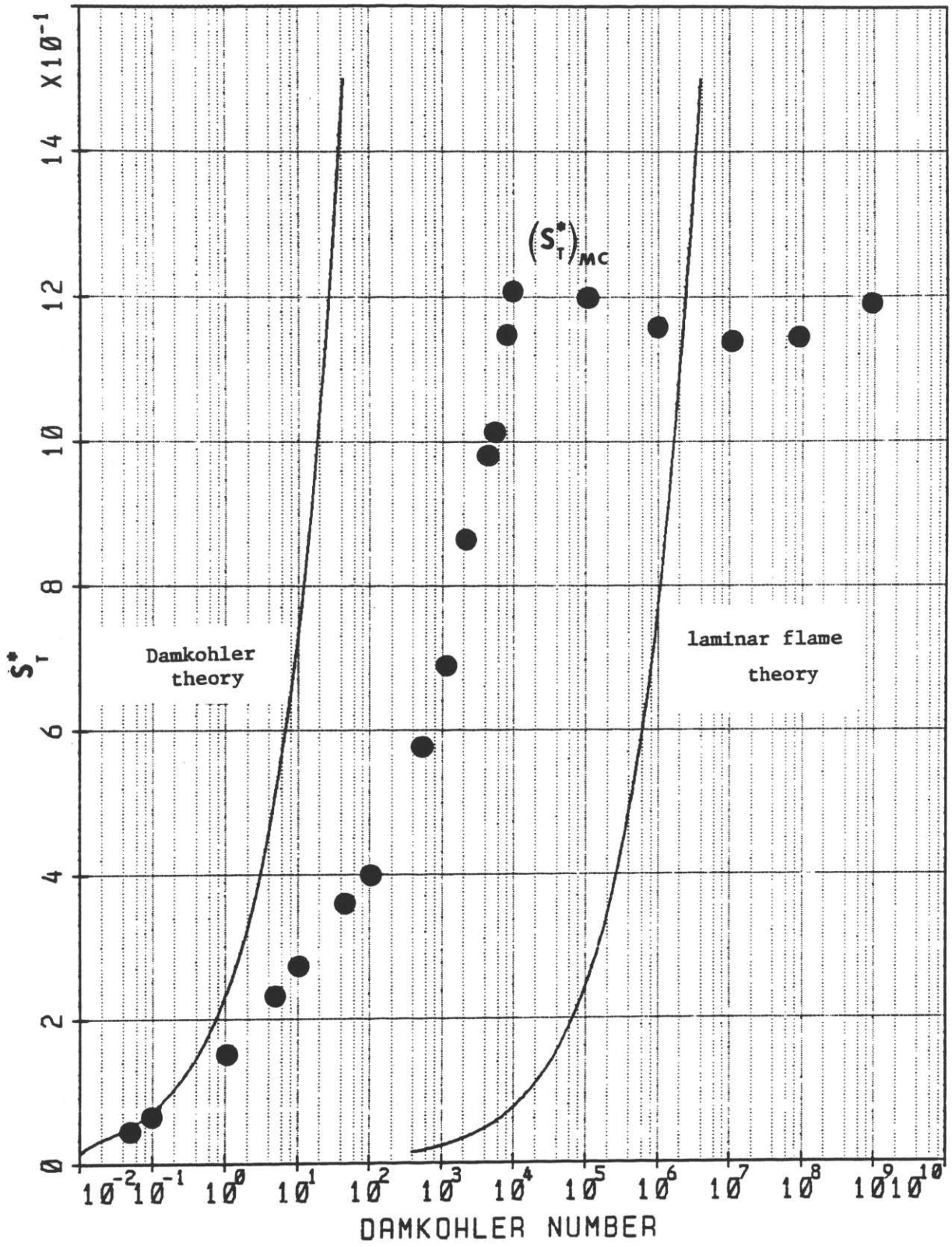

Figure 16 - Comparison of flame speed calculations from Monte Carlo model to theory

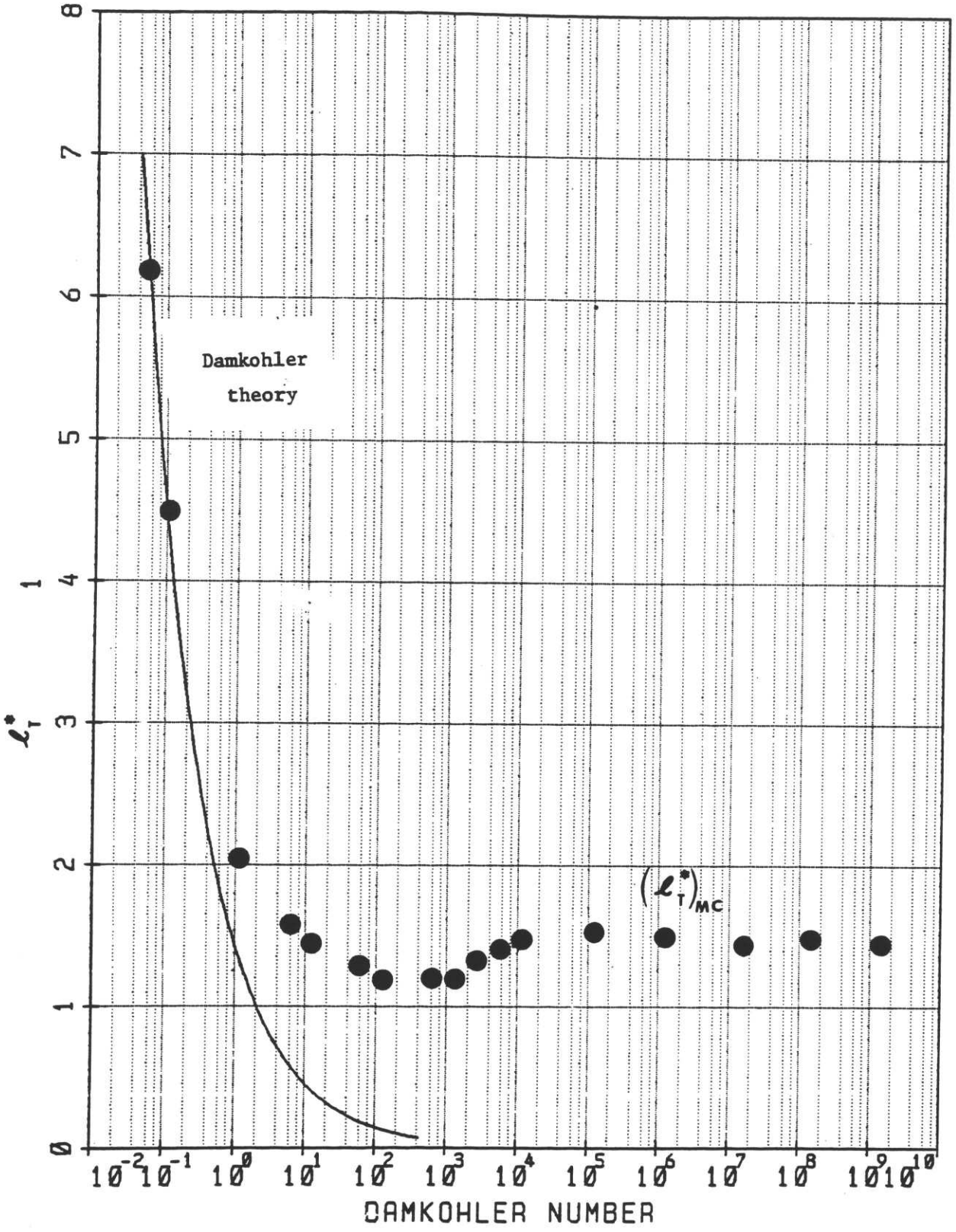

Figure 17 - Comparison of flame thickness calculations from Monte Carlo model to theory

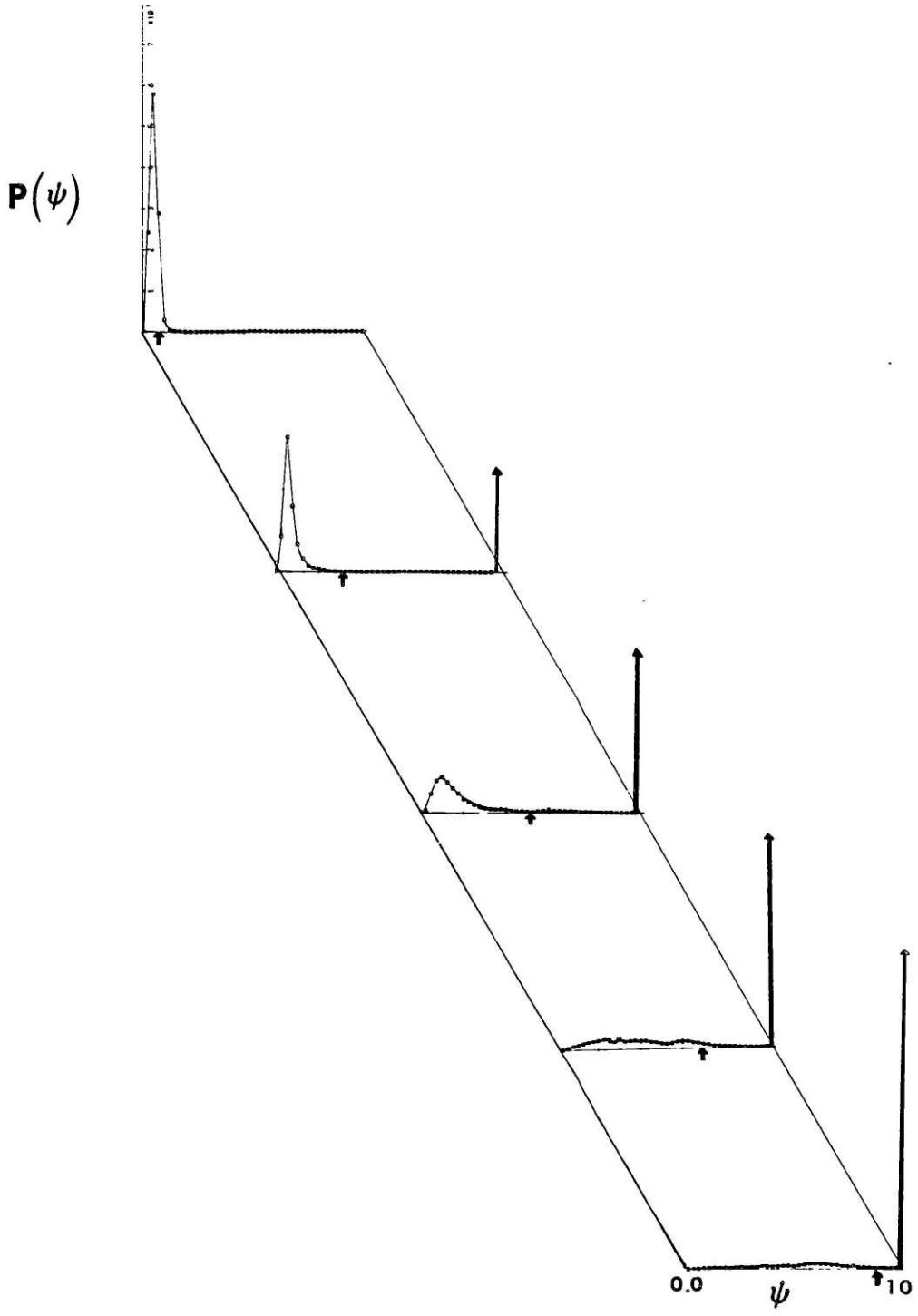

Figure 18 - Pdf's of C for Damkohler number of 10.0. Mean values of C are indicated by arrows.

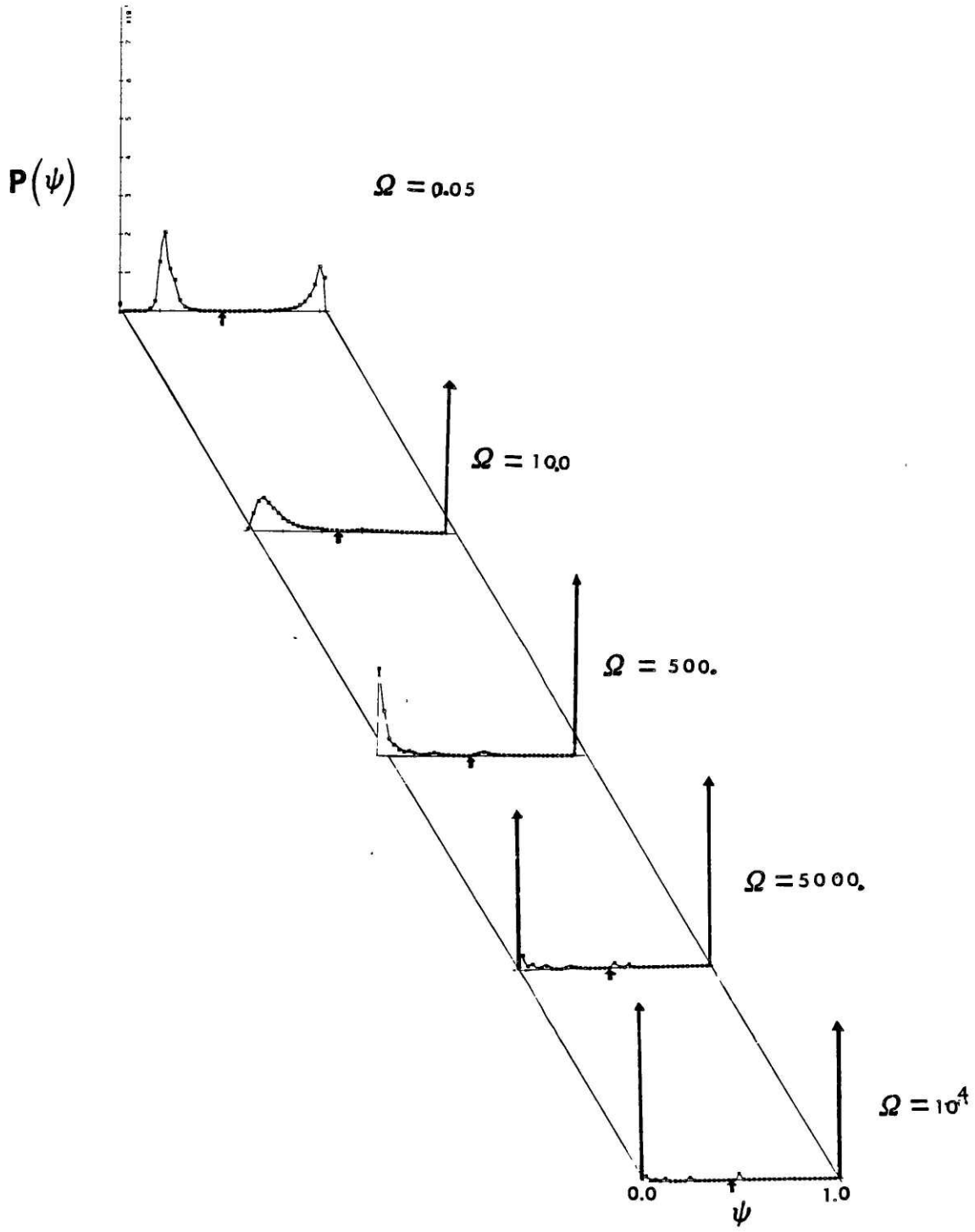

Figure 19 - Pdf's of C at $X^*_{0.5}$ for a range of Damkohler numbers

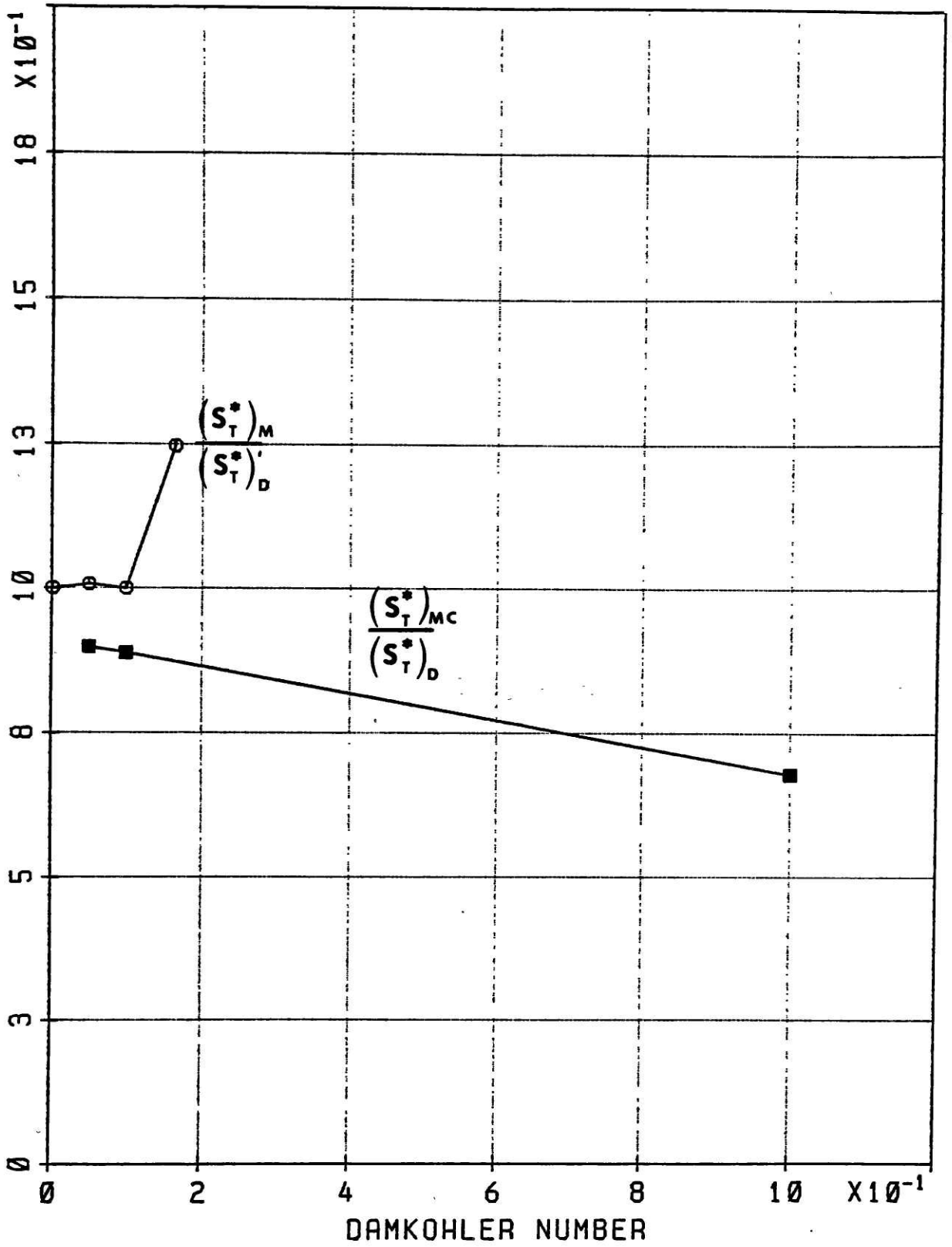

Figure 20 - Ratios of calculated flame speeds from the moment model and Mone Carlo model to calculated flame speeds from theory

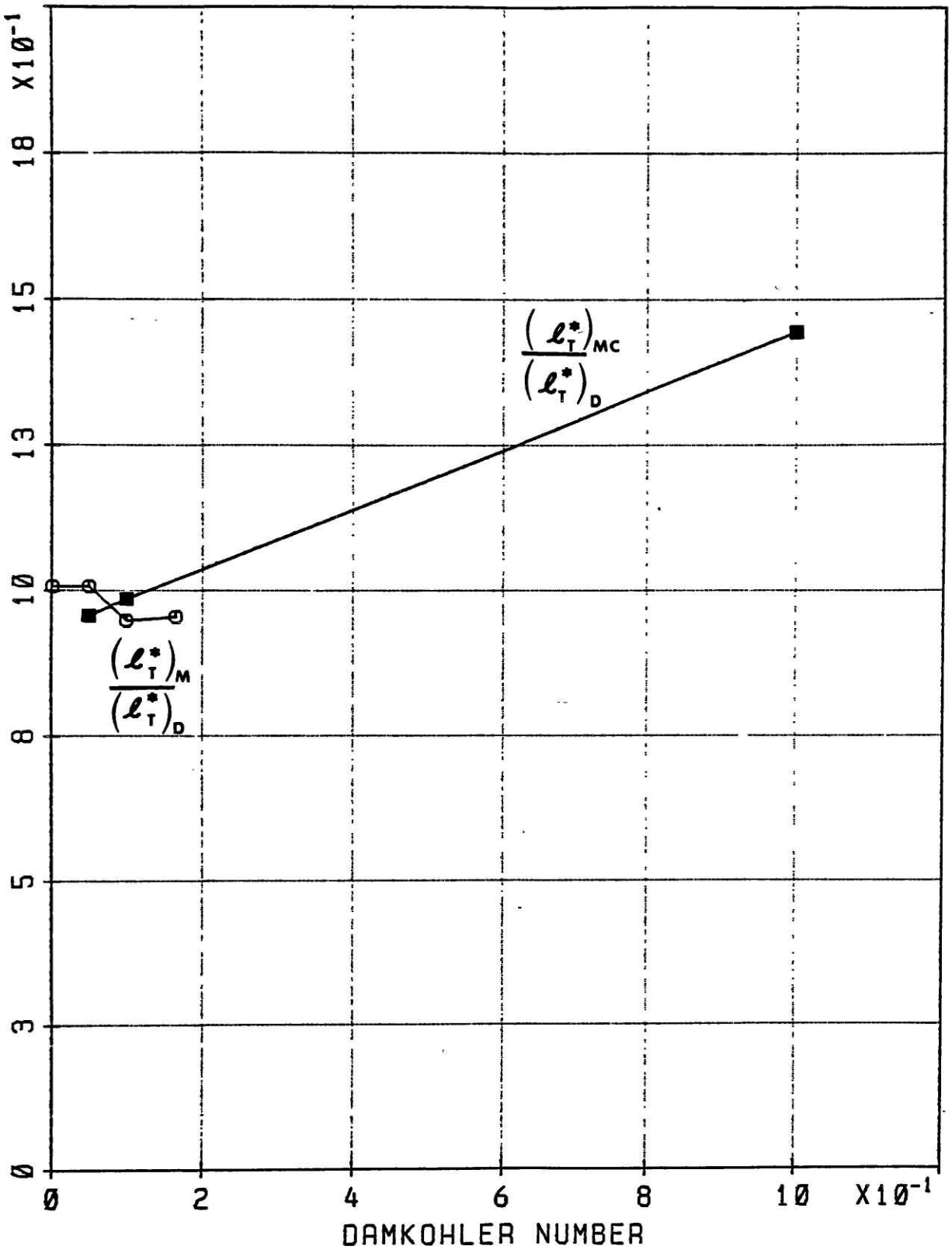

Figure 21 - Ratios of calculated flame thicknesses from the moment model and Monte Carlo model to calculated flame thicknesses from theory

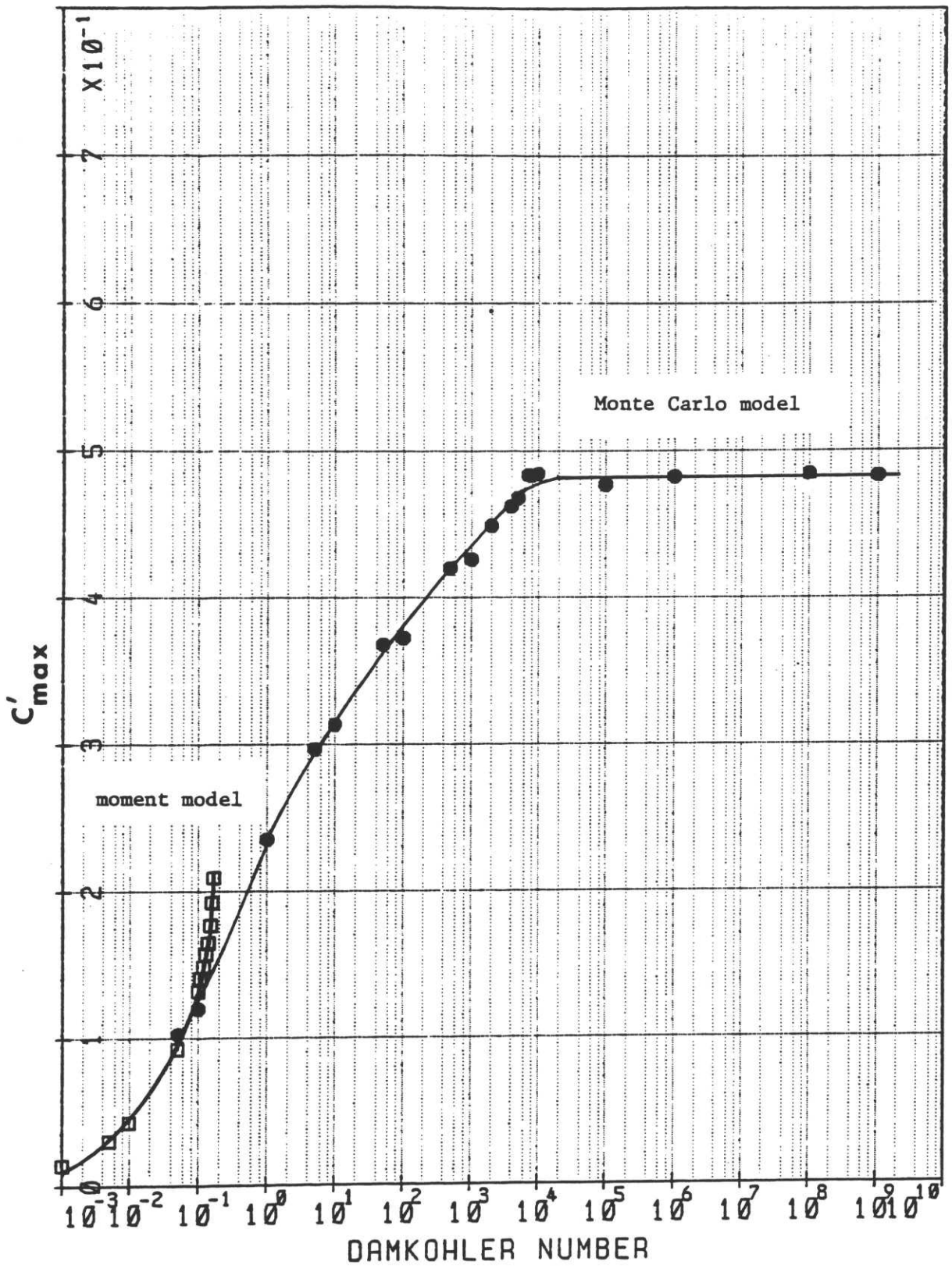

Figure 22 - Maximum values of C' from the moment model and the Monte Carlo model

Appendix A

A finite difference scheme is used to solve Equations (2.47) and (2.48). The following equations list some of the terms found in (2.47) and (2.48) and their finite difference expressions.

$$\frac{\partial \langle C \rangle}{\partial t^*} = \frac{\langle C_j^n \rangle - \langle C_j^{n-1} \rangle}{\Delta t^*} \tag{A1}$$

where $\langle C_j^n \rangle$ is the scalar concentration at node j and the current time, n.

$$\frac{\partial \langle C \rangle}{\partial X^*} = \frac{\langle C_j^n \rangle - \langle C_{j-1}^n \rangle}{\Delta X^*} \tag{A2}$$

The term $\langle C_j^n \rangle$ can be simplified as $\langle C_j \rangle$ with the n being understood.

$$\frac{\partial^2 \langle C \rangle}{\partial X^{*2}} = \frac{\langle C_{j+1} \rangle - 2\langle C_j \rangle + \langle C_{j-1} \rangle}{\Delta X^{*2}} \tag{A3}$$

$$\left\{\frac{\partial \langle C \rangle}{\partial X^*}\right\}^2 = \left\{\frac{1}{2}\left(\frac{\langle C_{j+1} \rangle - \langle C_j \rangle}{\Delta X^*}\right)^2 + \frac{1}{2}\left(\frac{\langle C_j \rangle - \langle C_{j-1} \rangle}{\Delta X^*}\right)^2\right\}^{n-1} \tag{A4}$$

$$C_\phi \langle C'^2 \rangle = C_\phi \langle C_j'^2 \rangle \tag{A5}$$

$$\frac{\partial s^*(\langle C \rangle)}{\partial C} \langle C'^2 \rangle = \frac{\partial s^*(\langle C_j^{n-1} \rangle)}{\partial C} \langle C_j'^2 \rangle \tag{A6}$$

$$\frac{\partial^2 s^*(\langle C \rangle)}{\partial C^2} \frac{\langle C'^2 \rangle}{2!} = \frac{\partial^2 s^*(\langle C_j^{n-1} \rangle)}{\partial C^2} \frac{\langle C_j'^2 \rangle}{2!} \tag{A7}$$

From these expressions, Equations (2.45) and (2.46) can be written in the form

$$\underline{\underline{A}}\ \underline{X} = \underline{B} \tag{A8}$$

where $\underline{X}$ is a column vector of the form

$$\underline{X} = \left[\langle C_1 \rangle, \langle C_1'^2 \rangle, \ldots, \langle C_j \rangle, \langle C_j'^2 \rangle, \ldots, \langle C_J \rangle, \langle C_J'^2 \rangle \right]^T \tag{A9}$$

and $\underline{\underline{A}}$ is a block tri-diagonal matrix composed of 2 x 2 matrices.

Therefore, Equations (2.47) and (2.48) can be solved simultaneously for $\langle C_j \rangle$ and $\langle C_j'^2 \rangle$, j=1,...,J using a Gaussian elimination routine which solves Equation (A8) for X. Initial conditions for $\langle C_j \rangle$ and $\langle C'^2_j \rangle$ are assumed and the criterion for convergence is Error $\leq 10^{-3}$ where

$$\text{Error} = \sum_{j=1}^{J} (B_j - A_j X_j)^2 \tag{A10}$$

Appendix B

This appendix contains the software listings of the computer programs used in this project. The following programs are listed:

```
C ------------------------------------------------------------
C
C MONTE CARLO MODEL
C
C THIS PROGRAM USES THE MONTE CARLO METHOD TO PREDICT
C FLAME SPEED AND FLAME THICKNESS OF A PREMIXED TURBULENT
C FLAME
C
C USAGE
C      INPUT FILE CONTAINING VALUE OF DAMKOHLER NUMBER
C       SUBROUTINE RXN
C
C      DESCRIPTION OF PARAMETERS
C
C       A-      2 X NCOUNT MATRIX USED TO CALCULATE
C               STANDARD DEVIATION OF FLAME THICKNESS
C               AND FLAME SPEED
C       C-     SCALAR CONCENTRATION OF CURRENT ELEMENT-
C               WHICH IS TRANSFERED TO  SUBROUTINE RXN
C    DAMK-      DAMKOHLER NUMBER
C    DELT-      TIME ELAPSED SINCE CURRENT ELEMENT LAST REACTED
C    DELX-     NONDIMESIONAL DISTANCE BETWEEM GRID NODES
C   DTDIF-     CHARACTERISTIC TIME STEP FOR DIFFUSION PROCESS
C   DTMIX-     CHARACTERISTIC TIME STEP FOR MIXING PROCESS
C       F-     ARRAY DIMENSIONED F(NV,NEL,NJ) WHICH
C                CONTAINS INFORMATION ABOUT C AND T
C    FAVG-     MEAN CONCENTRATION AT EACH NODE
C   FDIFF-     A CHECK WHICH SHOULD BE POSITIVE IF THE
C               FLUCTUATIONS ARE CORRECT
C   FLUCT-     FLUCTUTATION FROM THE MEAN AT EACH NODE
C    FREQ-     THE FREQUENCY WITH WHICH THE PDF
C               CALCULATIONS ARE PERFORMED
C  FSPEED-     INSTANTANEOUS FLAME SPEED
C    INIT-     TRANSFERED TO SUBROUTINE RXN SO THAT INPUT
C               FILE INTE.DAT IS READ ONCE
C  NDIFST-     NUMBER OF DIFFUSION STEPS PERFORMED
C      ND-     NUMBER OF ELEMENTS COMMUTED DURING DIFFUSION
C     NEL-     NUMBER OF ELEMENTS IN EACH ENSEMBLE
C   NGRID-     NUMBER OF BLOCKS IN PDF HISTOGRAM
C      NJ-     NUMBER OF GRID NODES, OR ENSEMBLES
C      NM-     NUMBER OF PAIRS MIXED DURING EACH TIME STEP
C  NMIXST-     NUMBER OF MIXING STEPS PERFORMED
C  NOSHIF-      NUMBER OF CONVECTION SHIFTS
C      NV-     NUMBER OF SPECIES
C    POUT-      ARRAY USED TO DETERMINE PDF
C      P1-     ARRAY USED TO DETERMINE PDF
C      P2-     ARRAY USED TO DETERMINE PDF
C     RAN-     SYSTEM RANDOM NUMBER GENERATOR
C     SD1-      STANDARD DEVIATION OF TBAR
C     SD2-     STANDARD DEVIATION OF SPBAR
C   SPBAR-      MEAN FLAME SPEED
C       T-     TIME
C     TAU-     INVERSE OF THE DAMKOHLER NUMBER, OR THE CHEMICAL
C               REACTION TIME
```

```
C    TBAR-      MEAN FLAME THICKNESS
C  TFINAL-     FINAL TIME VALUE
C   THICK-     INSTANTANEOUS FLAME THICKNESS
C     TSS-     PREDICTED VALUE OF T AT STEADY-STATE
C      XL-      NONDIMENSIONAL LENGTH OF THE GRID
C
C-----------------------------------------------------------------
C
      PARAMETER NV=2,NEL=200,NJ=51,NGRID=41
      VIRTUAL F(NV,NEL,NJ),A(2,200)
      DIMENSION FAVG(NJ),FLUCT(NJ),FDIFF(NJ)
      DIMENSION P1(NGRID),P2(NGRID),POUT(9,NGRID)
      COMMON/ILIST/INIT,C,DELT
C
C
C
C
C SET INITIAL VALUES
C
      DATA T/0./
C
C
C
C
C
      DATA NMIXST,NDIFST,NOSHIF,NCOUNT,IPDF,INIT/6*0/
C
C
      OPEN(UNIT=4,NAME='MONTE.DAT',FORM='FORMATTED',TYPE='OLD',
     & READONLY)
      READ(4,*)DAMK
      CLOSE(UNIT=4)
      TAU=1./DAMK
      TSTART=SECNDS(0.)
      FREQ=.1
      TSS=100.
      TFINAL=625. * (NGRID - 1)/FLOAT(NEL) * FREQ + TSS
C
      IY=INT(SECNDS(0.0)/60.)
      CALL IDATE (L,M,N)
C THE ABOVE LINES SUPPLY THE SEEDS FOR THE RANDOM NUMBER
C GENERATOR
C
      NM=NEL*.05
      ND=NEL * .1
C
      DELX=.2/(DAMK**.5)
      XL=DELX * (NJ-1)
      DTDIF=ND * DELX ** 2/(FLOAT(NEL) * .129)
      DTMIX= NM/(FLOAT(NEL) * 2.)
C
C
C INITIALIZE POUT ARRAY
C
```

```
      DO 111 J=1,NGRID
      DO 111 I=1,9
111   POUT(I,J)=0.
C
C
C
C
      OPEN(UNIT=4,NAME='MONTE.OUT',FORM='FORMATTED',TYPE='NEW')
      OPEN(UNIT=2,NAME='MONTE.TMP',FORM='FORMATTED',TYPE='NEW')
      CLOSE(UNIT=2)
      WRITE(4,995)TAU,DAMK
995   FORMAT(1X,'THE RXN RATE IS ',E10.3,/1X,
     &' DAMK.  = ',E10.3)
      WRITE(4,99)
99    FORMAT(1X,'THIS PROGRAM UTILIZES THE CURL MIXING MODEL
     & WITH REALISTIC RXN RATE')
      WRITE(4,*)IY,M
C INITIALIZE
C ASSUME STEP FUNCTION FOR INITIAL CONDITIONS
      NJ2=NJ/2
      NEW=NJ2 - 1
      NEW1=NEW + 1
C
3     DO 11 J=1,NEW
      DO 11 I=1,NEL
      F(2,I,J)=0.
11    F(1,I,J)=1.
      DO 1000 J=NEW1,NJ
      DO 1000 I=1,NEL
      DO 1000 N=1,NV
1000  F(N,I,J)=0.0
      XOLD=(NEW1+NEW)/2.
4     WRITE(4,2)DELX
2     FORMAT(1X,' DELX = ',F10.4)
      WRITE(4,1)XL,NJ,NEL
1     FORMAT(1X,' LENGTH IS ',F10.4,1X,'NJ IS',I4,
    & 1X,'NEL IS',I4)
      WRITE(4,5000)DTDIF,DTMIX
5000  FORMAT(1X,'DTDIF IS',F10.4,1X,'DTMIX IS',1X,F10.4)
C
C
      SHOLD=0.
      TOLD=0.
C
C
C
C
C CALL ROUTINES TO SIMULATE RXN,MIXING,DIFFUSION AND
C CONVECTION,DEPENDING ON CHRONOLOGICAL ORDER
C
5     IF(((T+DTDIF) .GT. TFINAL) .AND.
     &     ((T+DTMIX) .GT. TFINAL))GOTO 999
      TMIX= (NMIXST + 1) * DTMIX
      TDIF=(NDIFST + 1) * DTDIF
```

```
      IF((TDIF .NE. TMIX) .AND. (TDIF .LT. TMIX)) GOTO 6
      NMIXST=NMIXST + 1
      T=TMIX
C
C THIS ROUTINE SIMULATES MIXING BY SELECTING NM
C PAIRS OF ELEMENTS AT EACH NODE AT RANDOM AND
C REPLACING THEIR CONC. VALUES BY THE
C AVERAGE CONCENTRATIONS
C
      DO 6010 J=1,NJ
        DO 6010 I=1,NM
          I1=INT(NEL * RAN(IY,M)) + 1
          I2=INT(NEL * RAN(IY,M)) + 1
          C=F(1,I1,J)
          DELT=(T - F(2,I1,J))/TAU
C
C CALL SUBROUTINE TO PERFORM REACTION PROCESS
C SUBROUTINE MUST BE CALLED FOR EACH ELEMENT
C
          CALL RXN
          F(1,I1,J)=C
          F(2,I1,J)=T
          C=F(1,I2,J)
          DELT=(T - F(2,I2,J))/TAU
          CALL RXN
          F(1,I2,J)=C
          F(2,I2,J)=T
           F(1,I1,J)=.5 * (F(1,I1,J) + F(1,I2,J))
6010  F(1,I2,J)=F(1,I1,J)
      IF((TDIF - TMIX) .GT. .00001) GOTO 13
6     NDIFST=NDIFST + 1
      T=TDIF
C
C
C
C THIS ROUTINE SIMULATES DIFFUSION BY COMMUTING ND
C ELEMENTS CHOSEN AT RANDOM FIRST IN THE FORWARD
C DIRECTION THAN IN THE REVERSE DIRECTION.(FORWARD
C WAS CHOSEN AS BEING IN THE POSITIVE X-DIRECTION,
C FROM BURNT TO UNBURNT)
C
C
      ND2=ND*2
C
C SELECT 2 SETS OF ND ELEMENTS AT RANDOM AND PLACE AT TOP
C COORDINATES OF ELEMENT ARRAY
C
      DO 7020 J=1,NJ
        DO 7020 I=1,ND2
           ITOP=NEL - I + 1
      I1=INT(ITOP*RAN(IY,M)) + 1
```

```
            DO 7020 NSPEC=1,NV
             STORE=F(NSPEC,I1,J)
             F(NSPEC,I1,J)=F(NSPEC,ITOP,J)
             F(NSPEC,ITOP,J)=STORE
7020    CONTINUE
C
C FORWARD DIFFUSION
C COMMUTE ND ELEMENTS AT TOP OF ELEMENT ARRAY FROM
C NODE I TO NODE (I + 1)
C
       NJ1=NJ - 1
       DO 7030 N=1,ND
       NUMEL=NEL - N + 1
         DO 7030 I=1,NJ1
           NODEB=NJ - I
           NODEF= NODEB + 1
         DO 7030 NSPEC=1,NV
7030    F(NSPEC,NUMEL,NODEF)=F(NSPEC,NUMEL,NODEB)
C
C SET CONCENTRATION OF ELEMENTS AT FIRST NODE TO THAT OF B.C.'S
C
       NBEG=NEL - ND + 1
       DO 7040 J=NBEG,NEL
         F(2,J,1)=T
7040     F(1,J,1)= 1.0
C
C BACKWARD DIFFUSION (FROM UNBURNT TO BURNT)
C
       DO 7050 N=1,ND
       NUMEL= NEL - ND - N + 1
         DO 7050 I=NJ1,1,-1
           NODEB=NJ - I
           NODEF= NODEB + 1
         DO 7050 NSPEC=1,NV
7050    F(NSPEC,NUMEL,NODEB)=F(NSPEC,NUMEL,NODEF)
C
C SET CONCENTRATION ON ELEMENTS AT LAST NODE TO THAT OF B.C.'S.
C
C
       NBEG1=NEL - ND2 + 1
       NEND = NEL - ND
       DO 7060 J=NBEG1,NEND
         F(2,J,NJ)=T
7060       F(1,J,NJ)=0.0
7010    CONTINUE
C
C
C CHECK ON WHETHER CONVECTION CORRECTION IS NECCESSARY
C ( IF AVG CONC. OF ELEMENTS AT NODE NJ2 HAS REACHED 0.5)
C
```

```
13      SUM=0.
        DO 7 I=1,NEL
            C=F(1,I,NEW1)
            DELT=(T - F(2,I,NEW1))/TAU
            CALL RXN
            F(1,I,NEW1)=C
            F(2,I,NEW1)=T
7         SUM=SUM + C
        AVG=SUM/FLOAT(NEL)
        IF(AVG .LT. .5) GOTO 50
        NOSHIF=NOSHIF + 1
C
C
C
C
C THIS ROUTINE SIMULATES THE CONVECTION PROCESS BY MOVING
C THE ELEMENTS AT NODE 1 TO THE LAST NODAL COORDINATE AND
C SHIFTING ALL OF THE VALUES OF THE ELEMENTS ACCORDINGLY
C
          DO 8010 J=2,NJ
            NJ1=J - 1
            DO 8010 I=1,NEL
             DO 8010 NSPEC=1,NV
              F(NSPEC,I,NJ1)=F(NSPEC,I,J)
8010    CONTINUE
        DO 8020 I=1,NEL
          F(2,I,NJ)=T
8020    F(1,I,NJ)=0.0
C
C
C DETERMINE PDF AT EVERY .1 TIME STEP
C
50      IF((MOD(T,FREQ) .GT. .0001) .OR. (T .LT. TSS)) GOTO 5
C
C UPDATE RXN PROCESS OF ALL ELEMENTS
C
        DO 9010 J=1,NJ
          DO 9010 I=1,NEL
            C=F(1,I,J)
            DELT=(T - F(2,I,J))/TAU
            CALL RXN
            F(1,I,J)=C
            F(2,I,J)=T
9010    CONTINUE
C
C
C DETERMINE MEAN CONCENTRATION AT EACH NODE
C
        DO 200 J=1,NJ
          SUM=0.
          DO 100 I=1,NEL
              SUM=SUM + F(1,I,J)
```

```
100       CONTINUE
200     FAVG(J)=SUM/FLOAT(NEL)
C
        IX=1
        AMEAN=.9
601     DO 602 J=IX,NJ
        IF(FAVG(J) .GT. AMEAN) IX1=J
        IF(FAVG(J) .LT. AMEAN) GOTO 603
602     CONTINUE
603     IX2=J
        IX=IX2 + 1
        DO 605 N=1,NGRID
        P1(N)=0.
605     P2(N)=0.
        DO 606 I=1,NEL
        JBOX1=NINT(F(1,I,IX1) * (NGRID - 1) + 1.)
        JBOX2=NINT(F(1,I,IX2) * (NGRID - 1) + 1.)
        P1(JBOX1)=P1(JBOX1) + 1.
606     P2(JBOX2)=P2(JBOX2) + 1.
        WT1=1 - (FAVG(IX1) - AMEAN)/(FAVG(IX1) - FAVG(IX2))
        WT2=1 + (FAVG(IX2) - AMEAN)/(FAVG(IX1) - FAVG(IX2))
        MEAN=NINT(AMEAN * 10.)
        DO 607 J=1,NGRID
607     POUT(MEAN,J)=POUT(MEAN,J) +P1(J) *WT1 + P2(J) * WT2
        AMEAN=AMEAN - .1
        IF(AMEAN .GT. .01) GOTO 601
C
        IPDF=IPDF + 1
444     IF((MOD(T,.5) .GT. .0001) .OR. (T .LT. TSS)) GOTO 5
C
C
C THIS ROUTINE DOES THE CALCULATIONS TO PRODUCE THE
C OUTPUT FILE
C
C
C CALCULATE FLAME THICKNESS AS NODE WHERE AVG CONC. IS .9 TO
C WHERE AVG. CONC. IS .1
C FIND X COORDINATE WHERE CONC. IS .5 TO DETERMINE FLAME SPEED
C
C
C
        DO 222 J=1,NJ
          IF (FAVG(J) .GT. .9) IX1=J
        IF( FAVG(J) .LT. .9) GOTO 201
222     CONTINUE
201     IX2=J
        XLL=(.9 - FAVG(IX1)) * (IX2 - IX1)/(FAVG(IX2)-FAVG(IX1))
     & + IX1
        IXNEW=IX2 + 1
```

```
        DO 202 J=IXNEW,NJ
        IF(FAVG(J) .GT. .5) IX1=J
          IF(FAVG(J) .LT. .5)GOTO 204
202     CONTINUE
204     IX2=J
        XPT5=(.5 - FAVG(IX1)) * (IX2 - IX1)/(FAVG(IX2)
     & - FAVG(IX1)) + IX1
        IXNEW=IX2 + 1
        DO 300 J=IXNEW,NJ
        IF (FAVG(J) .GT. .1) IX1=J
        IF(FAVG(J) .LT. .1) GOTO 301
300     CONTINUE
301     IX2=J
        XR=(.1 - FAVG(IX1)) * (IX2- IX1)/(FAVG(IX2)
     & - FAVG(IX1)) + IX1
C
C
C CALCULATE FLAME THICKNESS
C
        THICK=(XR - XLL) * DELX
        WRITE(4,223) THICK,XLL,XR
223     FORMAT(1X,'FLAME THICKNESS IS ',E12.4,1X,
     & 'COORDS ARE ',F10.4,F10.4)
C
C CALCULATE FLAME SPEED
C
        DELSH=NOSHIF - SHOLD
        FSPEED=(XPT5 - XOLD + DELSH) * DELX/(T - TOLD)
        OPEN(UNIT=2,NAME='MONTE.TMP',FORM='FORMATTED',
     &TYPE='OLD',SHARED,ACCESS='APPEND')
        WRITE(2,*)THICK,FSPEED
        CLOSE(UNIT=2)
        XOLD=XPT5
        TOLD=T
        SHOLD=NOSHIF
        WRITE(4,224)FSPEED,T
224     FORMAT(1X,'FLAME SPEED IS ',F10.4,1X,' TIME IS ',F10.4)
        NCOUNT=NCOUNT + 1
        A(1,NCOUNT)=THICK
        A(2,NCOUNT)=FSPEED
C
C CALCULATE THE FLUCTUATION FROM THE MEAN
C
996     DO 500 J=1,NJ
        SUM= 0.0
          DO 700 I=1,NEL
700         SUM= SUM + (F(1,I,J) - FAVG(J))**2
          FLUCT(J)=(SUM/FLOAT(NEL))**.5
500     CONTINUE
        DO 600 J=1,NJ
600     FDIFF(J)=(FAVG(J) * (1. - FAVG(J)))**.5 -FLUCT(J)
        WRITE(4,1002)
1002    FORMAT(' F AVERAGE',2X,'FLUCTUATIONS',2X,'FLUC. CHECK')
        DO 1003 J=1,NJ
```

```
1003    WRITE(4,1001)FAVG(J),FLUCT(J),FDIFF(J)
1001    FORMAT(1X,F10.4,2X,F10.4,2X,F10.4)
        GOTO 5
999     WRITE(4,2000) NDIFST,NMIXST,NOSHIF
2000    FORMAT(' NDIFST IS',I5,' NMIXST IS ',
     & I5,1X,',NOSHIF IS ',I5)
        AVG=IPDF * NEL
        DO 888 J=9,1,-1
        AMEAN=J/10.
        WRITE(4,891)AMEAN
891     FORMAT(1X,' AVG. CONC.  IS ',F8.1)
892     DO 894 I=1,NGRID
894     POUT(J,I)=POUT(J,I)/AVG
        WRITE(4,893)(POUT(J,I),I=1,NGRID)
893     FORMAT(1X,10F12.4)
888     CONTINUE
C
C
C CALCULATE MEAN
C
        SUM1=0.
        SUM2=0.
        DO 991 J=1,NCOUNT
          SUM1=SUM1 + A(1,J)
991     SUM2=SUM2 + A(2,J)
        TBAR=SUM1/FLOAT(NCOUNT)
        SPBAR=SUM2/FLOAT(NCOUNT)
C
C CALCULATE STANDARD DEVIATION
C
        SUM1=0.
        SUM2=0.
        DO 992 J=1,NCOUNT
          SUM1=SUM1 + A(1,J)**2
992     SUM2=SUM2 + A(2,J)**2
        SD1=SQRT((SUM1 - NCOUNT * TBAR**2)/FLOAT(NCOUNT-1))
        SD2=SQRT((SUM2 - NCOUNT * SPBAR**2)/FLOAT(NCOUNT - 1))
        WRITE(4,997)TBAR,SD1,SPBAR,SD2
997     FORMAT(1X,'MEAN FLAME THICKNESS IS ',F10.4,
     &',ST.DEV. IS ',F10.4/',MEAN FLAME SPEED IS ',F10.4,
     &',ST.DEV. IS ',F10.4)
        TT=(SECNDS(0.)-TSTART)/60.
        WRITE(4,*)TT
        CLOSE(UNIT=4)
        STOP
        END
C
```

```
C-------------------------------------------------------
C
C       SUBROUTINE RXN
C
C       THIS SUBROUTINE SIMULATES THE MIXING BY INTERPOLATING
C       VALUES OBTAINED FROM THE INTEGRATION SCHEME, RUNGE
C
C       USAGE
C       CALL RXN
C       COMMON/ILIST/INIT,C,DELT
C
C       DESCRIPTION OF PARAMETERS
C
C         C-      MEAN CONCENTRATION OF CURRENT ELEMENT
C     DELT-       TIME SINCE LAST REACTION STEP OCCURED FOR
C                 CURRENT ELEMENT
C     INIT-       IF 0 READIN DATA FROM INTEGRATION OF SOURCE TERM
C                 SET TO 1 AFTER INITIAL READING OF DATA
C
C
C
C       INPUT FILES
C        INTE.DAT- CONTAINS DATA FROM FOURTH ORDER INTEGRATION
C                OF SOURCE TERM
C
C
C     METHOD
C
C       AN INPUT FILE CONTAINING THAT RESULT OF THE FOURTH ORDER
C       INTEGRATION OF THE SOURCE TERM IS NEEDED. FROM THE MEAN
C       CONCENTRATION OF THE CURRENT ELEMENT, THE CORRESPONDING
C       TIME IS FOUND USING LINEAR INTERPOLATION. THEN DELT
C       IS ADDED TO THAT TIME AND THE NEW, CORRESPONDING
C       CONCENTRATION IS FOUND USING LINEAR INTERPOLATION
C
C
C-------------------------------------------------------------
       SUBROUTINE RXN
C
C
       DIMENSION CT(502)
       COMMON/ILIST/INIT,C,DELT
       STEP=.002
       ITOP=502
C
C READ INPUT FILE
C
       IF(INIT .NE. 0) GOTO 10
       INIT =1
       OPEN(UNIT=3,NAME='INTE.DAT',FORM='FORMATTED',READONLY,
     & TYPE='OLD')
       READ(3,*)(CT(J),J=1,ITOP)
       CLOSE(UNIT=3)
C
```

```
C IF C IS LESS THAN .001 OR GREATER THAN .999 SUBROUTINE
C RETURNS
C
10      IF(C .LE. .001)RETURN
        IF(C .LT. .999) GOTO 20
        C=1.
        RETURN
C
C DETERMINE LOCATION IN ARRAY
C
20      I=(C + .001)/STEP + 1
        I1=I + 1
C
C CALCULATE VALUE OF TO USING LINEAR INTERPOLATION
C
        TO=CT(I) + (C - STEP*(I-1) + .001)*(CT(I1) - CT(I))/STEP
        T=TO + DELT
C
C SEARCH FOR LOCATION OF NEW VALUE OF T
C
        NSTEP2=ITOP/2
        J=I + (ITOP - I)/2
        DO 45 K=1,NSTEP2
          ISUM=NINT(FLOAT(ITOP - J)/2.)
          IF(CT(J) .LT. T) GOTO 46
          ISIGN=-1
          ITOP=J
          GOTO  44
46        ISIGN=1
44        J=J + ISUM * ISIGN
        IF(ISUM .EQ. 1)GOTO 40
45      CONTINUE
40      IF(CT(J) .GT. T) J=J - 1
        J1=J + 1
C
C CALCULATE NEW CONCENTRATION USING LINEAR INTERPOLATION
C
        C=STEP * (J - 1.) - .001 + (T - CT(J)) *
     & STEP/(CT(J1) - CT(J))
        C=AMIN1(C,1.)
        RETURN
        END
```

```
C-------------------------------------------------------------
C
C MAIN PROGRAM
C
C THIS PROGRAM SOLVES THE SCALAR EQUATION USING THE
C DAMKOHLER LIMIT AND ASSUMES GRADIENT DIFFUSION
C WITH SCALAR FLUCTUATIONS.
C
C DESCRIPTION OF PARAMETERS:
C
C       NJ-     NUMBER OF GRID NODES
C        F-     AN ARRAY CONTAINING CHANGES IN <C> AND <C'**2>
C        A-
C        B-     THESE ARE ARRAYS WHICH COMPOSE THE TRI-DIAGONAL
C        C-     MATRIX FORMED FROM THE FINITE DIFFERENCE METHOD
C
C        D-     A NULL ARRAY USED BY THE MATRIX INVERSION
C               SUBROUTINE
C     FOLD-     CONTAINS THE LAST VALUES OF <C> AND <C'**2>
C        G-     CONTAINS THE OLD VALUES FROM F
C       C1-     THE VALUE OF <C> AT THE CURRENT GRID NODE
C    S(C1)-     THE SOURCE TERM EVALUATED AT <C>=C1
C   DS(C1)-     THE FIRST DERIVATIVE OF THE SOURCE TERM
C               EVALUATED AT <C>=C1
C DS2(C1)-      THE SECOND DERIVATIVE OF THE SOURCE TERM
C               EVALUATED AT <C>=C1
C    DELX-      THE SPACING BETWEEN GRID NODES,CALCULATED USING
C                  DELX=X*/(DAMK**.5), WHERE X*=.03
C       XL-     NONDIMENSIONAL LENGTH OF GRID,DELX*(NJ-1)
C     DAMK-     DAMKOHLER NUMBER
C       DT-     TIME STEP CALCULATED USING DT=DT*/DAMK
C               DT*=.025
C    XMEAN-     LOCATION WHERE <C>=0.5
C       SL-     FLAME SPEED
C    THICK-     FLAME THICKNESS
C        T-     CURRENT TIME
C   TFINAL-     FINAL TIME
C      CMU-     A CONSTANT FROM TURBULENCE THEORY,CMU=0.09
C   SIGPHI-     A CONSTANT FROM TURBULENCE THEORY,SIGPHI=0.7
C
C
C
C ROUTINES NEEDED
C
C       MATRIX
C
C INPUT FILES NEEDED
C
C       PARAM.DAT: THIS FILE CONTAINS THE DAMKOHLER NUMBER
C       CONC.DAT: THIS FILE CONTAINS <C> AND <C'**2> FROM
C                 A PREVIOUS RUN TO BE USED AS INITIAL
C                 CONDITIONS FOR CURRENT RUN
C
C METHOD
```

```
C       THE ROUTINE SOLVES TWO SIMULTANEOUS PARTIAL DIFFERENTIAL
C       EQUATIONS FOR <C> AND <C'**2>. INPUTS CONSIST OF THE
C       DAMKOHLER NUMBER, GEOMETRY SPECIFICATIONS, AND TIME
C        STEP.THE OUTPUT FROM THE PROGRAM IS A LIST OF THE
C        ITERATIONS OF FLAME THICKNESS, FLAME SPEED, ERROR FOR
C        EACH EQUATION,AND FINAL VALUES OF <C> AND <C'**2>.
C        AFTER AN INITIAL RUN,THE RESULTS FROM THAT RUN MAY
C        USED FOR INITIAL CONDITIONS FOR SEQUENTIAL RUNS.
C
C-----------------------------------------------------------
C
C
        PARAMETER NJ=301
        VIRTUAL F(2,NJ),A(2,2,NJ),B(2,2,NJ),C(2,2,NJ),
     &             D(2,2,NJ),FOLD(2,NJ),G(2,NJ)
C
        S(C1)=6.11E07 * C1*(1.-C1)*EXP(-100./(1+6.*C1))
        DS(C1)=6.11E07*EXP(-100./(1.+6.*C1))*(1.-2.*C1 +
     &          C1*(1.-C1)*600./((1.+6.*C1)**2))
        DS2(C1)=6.11E07*EXP(-100./(1.+6.*C1))*(600./((1.+6.*C1)
     &**2)*(2. - 4.*C1 +C1*(1.-C1)*600./((1.+6.*C1)**2)
     & - C1*(1.-C1)*12./(1.+6.*C1)) -2.)
C
C
        DATA SL,TFINAL,T/1.,10000.,0./
        DATA CMU,SIGPHI,CPHI,DAMK/.09,.7,2.,.001/
        OPEN(UNIT=4,NAME='FLUCTV.OUT',FORM='FORMATTED',
     & TYPE='NEW')
        OPEN(UNIT=2,NAME='FLUCTV.DAT',FORM='FORMATTED',
     & TYPE='NEW')
        OPEN(UNIT=3,NAME='FLUCT.OUT',FORM='FORMATTED',
     & TYPE='NEW')
        OPEN(UNIT=1,NAME='PARAM.DAT',FORM='FORMATTED',
     & TYPE='OLD',READONLY)
        CALL ERRSET(72,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
        CALL ERRSET(73,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
        READ(1,*)DAMK
        CLOSE(UNIT=1)
        WRITE(4,111)
111     FORMAT(1X,' THIS PROGRAM SOLVES THE MEAN SCALAR
     & EQTN. W/ FLUCT.')
        WRITE(4,1)SL,NJ
1       FORMAT(1X,' INITIAL FLAME SPEED =  ',F10.4/,
     & ' GRID SIZE = ',I4)
C
C CALC. DELX.,DT,USING NORMALIZED DELX* AND DT*
C
        DELX=.03/(DAMK**.5)
        XL=DELX*(NJ-1)
        DT=.025/DAMK
C
        WRITE(4,3)XL,DELX,DT
3       FORMAT(1X,' LENGTH = ',F10.4,'   DELX =',F10.4/,
     & ' TIME STEP = ',E12.4)
```

```
      WRITE(4,555)DAMK
555   FORMAT(1X,' DAMKOHLER NUMBER = ',E12.4)
      WRITE(4,223)
223   FORMAT(1X/,' FLAME THICKNESS',4X,' FLAME SPEED',4X,
     & ' FLUCTUATIONS'/)
C
C
      CLOSE(UNIT=3)
      CI=-CMU/(SIGPHI * DELX**2)
      NJ1=NJ - 1
C
C DEFINE FLAME FRONT
C ************************************************************
C NOTE:THIS SECTION IS USED FOR INITIAL RUN AND THEN COMMENTED
C OUT AND REPLACED BY THE FOLLOWING SECTION WHICH READS IN THE
C RESULTS FROM A PREVIOUS RUN
C ************************************************************
C     DO 20 J=1,NJ
C20     FOLD(2,J)=0.
C     NJ4=NJ/4
C     NJ43=NJ4 * 3
C     NJDIF=NJ43 - NJ4
C     DO 40 J=1,NJ4
C40     FOLD(1,J)=0.
C     DO 21 J=NJ4,NJ43
C21   FOLD(1,J)=1. + SIN(3.14159*((J-NJ4)/(2. * NJDIF) - .5))
C     DO 41 J=NJ43,NJ
C41     FOLD(1,J)=1.
C     DO 2002 J=NJ4,NJ
C
C     IF(FOLD(1,J) .LT. .5) IX1=J
C       IF(FOLD(1,J) .GT. .5)GOTO 2004
C2002 CONTINUE
C2004 IX2=J
C     XMEAN=(.5-FOLD(1,IX1)) * (IX2 - IX1)/(FOLD(1,IX2) -
C    &FOLD(1,IX1)) + IX1
C ************************************************************
C NOTE: THIS SECTION IS USED ONLY AFTER AN INITIAL RUN HAS BEEN
C MADE READ INITIAL CONDITIONS FROM DATA FILE AND CALC. WHERE
C THE MEAN CONC. IS TO CALC. FLAME SPEED.
      OPEN(UNIT=1,NAME='CONC.DAT',FORM='FORMATTED',TYPE='OLD',
     & READONLY)
      DO 40 I=1,2
40    READ(1,*)(FOLD(I,J),J=1,301)
      DO 41 J=1,301
41    FOLD(2,J)=ABS(FOLD(2,J))
      CLOSE(UNIT=1)
C************************************************************
C
C DETERMINE WHERE <C> IS 0.5
C
      DO 2002 J=1,NJ
      IF(FOLD(1,J) .LT. .5) IX1=J
        IF(FOLD(1,J) .GT. .5)GOTO 2004
```

```
2002    CONTINUE
2004    IX2=J
        XMEAN=(.5-FOLD(1,IX1)) * (IX2 - IX1)/(FOLD(1,IX2) -
     & FOLD(1,IX1)) + IX1
5       T=T + DT
        IF((SL .LT. 0.) .AND. (T .LT. (5. * DT)))SL=1.
        IF(T .GT. TFINAL) GOTO 999
C
C INITIALIZE D
C
        DO 65 J=1,NJ
          DO 65 I=1,2
            DO 65 L=1,2
65      D(L,I,J)=0.
C
C SET BOUNDARY CONDITIONS
C
        DO 6 L=1,NJ,NJ1
          DO 6 I=1,2
            DO 7 J=1,2
              A(I,J,L)=0.
              B(I,J,L)=0.
7           C(I,J,L)=0.
            B(I,I,L)=1.
            F(I,L)=0.
6       FOLD(I,L)=0.
C       F(1,NJ)=1.
        FOLD(1,NJ)=1.
C
C
C DETERMINE A,B,C AND RESIDUE AT EACH GRID NODE
C
        DO 30 J=2,NJ1
        C1=FOLD(1,J)
        A(1,1,J)=CI - SL/DELX
        A(1,2,J)=0.
        A(2,1,J)=0.
        A(2,2,J)=A(1,1,J)
C
C
        B(1,1,J)=1/DT + SL/DELX - 2. * CI
        B(1,2,J)=-.5 * DAMK *DS2(C1)
        B(2,1,J)=0.
        B(2,2,J)=B(1,1,J) + CPHI - 2.*DAMK*DS(C1)
C
C
        C(1,1,J)=CI
        C(1,2,J)=0.
        C(2,1,J)=0.
        C(2,2,J)=CI
C
C
        R1=DAMK*S(C1) + C1/DT
        R2=-CI * ((FOLD(1,J+1)-C1)**2 + (C1-FOLD(1,J-1))**2)
```

```
     & + FOLD(2,J)/DT
       F(1,J)=R1- A(1,1,J)*FOLD(1,J-1) - A(1,2,J)*FOLD(2,J-1)
     &         - B(1,1,J)*FOLD(1,J) - B(1,2,J)*FOLD(2,J)
     &         - C(1,1,J)*FOLD(1,J+1) - C(1,2,J)*FOLD(2,J+1)
       F(2,J)=R2- A(2,1,J)*FOLD(1,J-1) - A(2,2,J)*FOLD(2,J-1)
     &         - B(2,1,J)*FOLD(1,J) - B(2,2,J)*FOLD(2,J)
     &         - C(2,1,J)*FOLD(1,J+1) - C(2,2,J)*FOLD(2,J+1)
       A(2,1,J)=A(2,1,J) - CI*2.*(FOLD(1,J)-FOLD(1,J-1))
       B(2,1,J)=B(2,1,J) + CI*2.*(2.*FOLD(1,J) - FOLD(1,J+1)
     &                   - FOLD(1,J-1))
       C(2,1,J)=C(2,1,J) + CI*2.*(FOLD(1,J+1) - FOLD(1,J))
       G(1,J)=F(1,J)
       G(2,J)=F(2,J)
30     CONTINUE
       CALL MATRIX(A,B,C,D,F,NJ)
C
       F(1,1)=-FOLD(1,1)
       F(1,NJ)=1. - FOLD(1,NJ)
       F(2,1)=-FOLD(2,1)
       F(2,NJ)=-FOLD(2,NJ)
C
C SOLVE FOR NEW VALUES OF <C> AND <C'**2>
C
       DO 31 J=1,NJ
         DO 31 I=1,2
31     FOLD(I,J)=FOLD(I,J) + F(I,J)
C
C FIND LOCATIONS ON GRID WHERE <C>=0.1,0.5,0.9
C
       DO 222 J=1,NJ
         IF (FOLD(1,J) .LT. .1) IX1=J
       IF( FOLD(1,J) .GT. .1) GOTO 201
222    CONTINUE
201    IX2=J
       XLL=(.1 - FOLD(1,IX1)) * (IX2 - IX1)/
     & (FOLD(1,IX2)-FOLD(1,IX1)) + IX1
       IXNEW=IX2 + 1
       DO 202 J=IXNEW,NJ
       IF(FOLD(1,J) .LT. .5) IX1=J
         IF(FOLD(1,J) .GT. .5)GOTO 204
202    CONTINUE
204    IX2=J
       XPT5=(.5-FOLD(1,IX1)) * (IX2 - IX1)/
     & (FOLD(1,IX2) - FOLD(1,IX1)) + IX1
       IXNEW=IX2 + 1
       DO 955 J=IXNEW,NJ
       IF (FOLD(1,J) .LT. .9) IX1=J
       IF(FOLD(1,J) .GT. .9) GOTO 301
955    CONTINUE
301    IX2=J
       XR=(.9 - FOLD(1,IX1)) * (IX2- IX1)/
     & (FOLD(1,IX2) - FOLD(1,IX1)) + IX1
C
C
```

```
C CALCULATE FLAME THICKNESS
C
      THICK=(XR - XLL) * DELX
C
C CALCULATE FLAME SPEED
C
      XDIFF=XMEAN - XPT5
      XMEAN=XPT5
C
C DETERMINE ERROR
C
      SL=SL + XDIFF * DELX/DT
      SUM1=0.
      SUM2=0.
      DO 50 J=1,NJ
      SUM1=SUM1 + (G(1,J)/DAMK)**2
      SUM2=SUM2 + (G(2,J)/DAMK)**2
50    CONTINUE
      OPEN(UNIT=3,NAME='FLUCT.OUT',FORM='FORMATTED',
     & TYPE='OLD',SHARED,ACCESS='APPEND')
      WRITE(4,115)THICK,SL,SUM1,SUM2
      WRITE(3,115)THICK,SL,SUM1,SUM2
      CLOSE(UNIT=3)
115   FORMAT(1X,4E12.4)
C
C CHECK WHETHER CONVERGENCE CRITERION IS MET
C
      IF((ABS(SUM1) .GT. .001)  .OR.
     & (ABS(SUM2) .GT. .001))GOTO 5
999   WRITE(2,*)(FOLD(1,I),I=1,NJ)
      WRITE(2,*)(FOLD(2,I),I=1,NJ)
      WRITE(4,*)(FOLD(1,I),I=1,NJ)
      WRITE(4,*)(FOLD(2,I),I=1,NJ)
      CLOSE(UNIT=4)
      CLOSE(UNIT=2)
      STOP
      END
```

```
      SUBROUTINE MATRIX(A,B,C,D,F,N)
C
C     THIS SUBROUTINE SOLVES A BLOCK TRIDIAGONAL SYSTEM USING
C      GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING.  THE FORM
C      OF THE SYSTEM IS AS FOLLOWS :
C
C   B(1)  C(1)  E(1)                                  X(1)      F(1)
C   A(2)  B(2)  C(2)  D(2)                            X(2)      F(2)
C         A(3)  B(3)  C(3)                            X(3)      F(3)
C         ..............                            ....      ....
C            ..............                   x     ....   =  ....
C            ..............                         ....      ....
C          A(N-2)  B(N-2)  C(N-2)  D(N-2)           X(N-2)    F(N-2)
C                A(N-1)  B(N-1)  C(N-1)             X(N-1)    F(N-1)
C                E(N)    A(N)    B(N)               X(N)      F(N)
C
C     WHERE A, B, C, D, AND E ARE KxK BLOCKS, X AND F ARE K
B COLUMN VECTORS.  D BLOCKS ARE USED IN PARTIAL PIVOTING WHICH
C REQUIRES D(2), D(3), ..., D(N-2) TO BE ZERO WHEN BEGINNING
C COMPUTATION.E(1) AND E(N) ARE REPLACED BY D(1) AND D(N),
C RESPECTIVELY.       ALSO, F IS SUBSTITUTED FOR X TO GET THE
C FINAL SOLUTION IN F.THUS, INPUTS TO THE SUBROUTINE ARE
C A, B, C, D, F, K, KD, AND N WHERE A, B, C, AND D ARE
C EXPLAINED ABOVE, F AS AN INPUT CONTAINS KNOWN TERMS, K IS
C THE DIMENSION OF BLOCKS, KD IS THE DIMENSION OF K, AND N IS
C THE NUMBER OF BLOCK ROWS. OUTPUT OF THE SUBROUTINE, WHICH IS
C THE FINAL SOLUTION, IS  RETURNED IN F.
C
C WRITTEN BY B.MINAIE
C
C
      PARAMETER KD=2,K=2
      DIMENSION A(KD,KD,1),B(KD,KD,1),C(KD,KD,1),D(KD,KD,1),
     & F(KD,1)
C
      DATA ZRO /1.E-10/
C
      NM1=N-1
      NM2=N-2
      DO 270 M=1,N
      DO 270 JJ=1,K
      MM1=M-1
      MP1=M+1
      MP2=M+2
C
C****  FIND LARGEST ELEMENT IN ABSOLUTE VALUE IN BLOCK B(M),
C M=1,...,N
C
      BMX=0.0
      DO 10 I=JJ,K
      IF (ABS(B(I,JJ,M)) .LT. BMX) GO TO 10
```

```
      BMX=ABS(B(I,JJ,M))
      IB=I
10    CONTINUE
      IF (M .EQ. N) GO TO 50
C
C****  FIND LARGEST ELEMENT IN ABSOLUTE VALUE IN BLOCK A(M+1)
C
      AMX=0.0
      DO 20 I=1,K
      IF (ABS(A(I,JJ,MP1)) .LT. AMX) GO TO 20
      AMX=ABS(A(I,JJ,MP1))
      IA=I
20    CONTINUE
C
C****  WHEN M=N-2, FIND LARGEST ELEMENT IN ABSOLUTE VALUE IN
C BLOCK D(N)
C
      IF (M .NE. NM2) GO TO 40
      DMX=0.0
      DO 30 I=1,K
      IF (ABS(D(I,JJ,MP2)) .LT. DMX) GO TO 30
      DMX=ABS(D(I,JJ,MP2))
      ID=I
30    CONTINUE
      IF (DMX .GT. AMX  .AND.  DMX .GT. BMX) GO TO 120
C
C--   CHECK FOR PIVOT IN A
C
40    IF (AMX .GT. BMX) GO TO 90
C
C****  COMMUTE ROWS IF OVERALL MAX. OCCURS IN BLOCK B(M)
C
50    IF (BMX .LE. ZRO) GO TO 340
      IF (IB .EQ. JJ) GO TO 150
C
C--   COMMUTE WITHIN B(M)
C
      DO 60 J=JJ,K
      BTMP=B(JJ,J,M)
      B(JJ,J,M)=B(IB,J,M)
60    B(IB,J,M)=BTMP
      IF (M .EQ. N) GO TO 80
C
C--   COMMUTE WITHIN C(M) FOR M .LT. N
C
      DO 70 J=1,K
      CTMP=C(JJ,J,M)
      C(JJ,J,M)=C(IB,J,M)
70    C(IB,J,M)=CTMP
C
C--   COMMUTE WITHIN D(M) FOR M .LT. N-1
```

```
C
      IF (M .GE. NM1) GO TO 74
      DO 72 J=1,K
      DTMP=D(JJ,J,M)
      D(JJ,J,M)=D(IB,J,M)
72    D(IB,J,M)=DTMP
74    CONTINUE
C
C--   COMMUTE WITHIN F(M)
C
80    FTMP=F(JJ,M)
      F(JJ,M)=F(IB,M)
      F(IB,M)=FTMP
      GO TO 150
90    CONTINUE
C
C**** COMMUTE ROWS IF OVERALL MAX. OCCURS IN BLOCK A(M+1)
C
      IF (AMX .LE. ZRO) GO TO 340
C
C--   COMMUTE BETWEEN B(M) AND A(M+1)
C
      DO 100 J=JJ,K
      BTMP=B(JJ,J,M)
      B(JJ,J,M)=A(IA,J,MP1)
100   A(IA,J,MP1)=BTMP
C
C--   COMMUTE BETWEEN C(M) AND B(M+1)
C
      DO 110 J=1,K
      CTMP=C(JJ,J,M)
      C(JJ,J,M)=B(IA,J,MP1)
110   B(IA,J,MP1)=CTMP
C
C--   COMMUTE BETWEEN D(M) AND C(M+1) FOR M .LT. N-1
C
3     IF (M .GE. NM1) GO TO 114
      DO 112 J=1,K
      DTMP=D(JJ,J,M)
      D(JJ,J,M)=C(IA,J,MP1)
112   C(IA,J,MP1)=DTMP
114   CONTINUE
C
C--   COMMUTE BETWEEN F(M) AND F(M+1)
C
      FTMP=F(JJ,M)
      F(JJ,M)=F(IA,MP1)
      F(IA,MP1)=FTMP
      GO TO 150
C
C**** WHEN M=N-2, COMMUTE ROWS IF OVERALL MAX. OCCURS IN
```

```
C BLOCK D(N)
C
120     IF (DMX .LE. ZRO) GO TO 340
C
C--     COMMUTE BETWEEN B(M) AND D(M+2)
C
        DO 130 J=JJ,K
        BTMP=B(JJ,J,M)
        B(JJ,J,M)=D(ID,J,MP2)
130     D(ID,J,MP2)=BTMP
C
        DO 140 J=1,K
C
C--     COMMUTE BETWEEN C(M) AND A(M+2)
C
        CTMP=C(JJ,J,M)
        C(JJ,J,M)=A(ID,J,MP2)
        A(ID,J,MP2)=CTMP
C
C--     COMMUTE BETWEEN D(M) AND B(M+2)
C
        DTMP=D(JJ,J,M)
        D(JJ,J,M)=B(ID,J,MP2)
        B(ID,J,MP2)=DTMP
140     CONTINUE
C
C--     COMMUTE BETWEEN F(M) AND F(M+2)
C
        FTMP=F(JJ,M)
        F(JJ,M)=F(ID,MP2)
        F(ID,MP2)=FTMP
150     CONTINUE
C
C****   ELIMINATE ELEMENTS IN BLOCK B(M), M=1,...,N
C
        JJP1=JJ+1
        IF (JJ .EQ. K) GO TO 200
        DO 190 I=JJP1,K
        BFACT=B(I,JJ,M)/B(JJ,JJ,M)
C
C--     COMPUTE NEW VALUES IN B(M)
C
        DO 160 J=JJ,K
        B(I,J,M)=B(I,J,M)-BFACT*B(JJ,J,M)
160     CONTINUE
C
C--     COMPUTE NEW VALUES IN C(M) FOR M .LT. N
C
        IF (M .EQ. N) GO TO 180
C
        DO 170 J=1,K
```

```
170     C(I,J,M)=C(I,J,M)-BFACT*C(JJ,J,M)
C
C--     COMPUTE NEW VALUES IN D(M) FOR M .LT. N-1
C
        IF (M .GE. NM1) GO TO 174
        DO 172 J=1,K
172     D(I,J,M)=D(I,J,M)-BFACT*D(JJ,J,M)
174     CONTINUE
C
C--     COMPUTE NEW VALUES IN F(M)
C
180     F(I,M)=F(I,M)-BFACT*F(JJ,M)
190     CONTINUE
C
C****   ELIMINATE ELEMENTS IN BLOCK A(M+1) FOR M .LT. N
C
200     IF (M .EQ. N) GO TO 270
        DO 230 I=1,K
        AFACT=A(I,JJ,MP1)/B(JJ,JJ,M)
C
C--     COMPUTE NEW VALUES IN A(M+1)
C
        DO 210 J=JJ,K
        A(I,J,MP1)=A(I,J,MP1)-AFACT*B(JJ,J,M)
210     CONTINUE
C
C--     COMPUTE NEW VALUES IN B(M+1) AND C(M+1)
C
        DO 220 J=1,K
        B(I,J,MP1)=B(I,J,MP1)-AFACT*C(JJ,J,M)
        C(I,J,MP1)=C(I,J,MP1)-AFACT*D(JJ,J,M)
220     CONTINUE
C
C--     COMPUTE NEW VALUES IN F(M+1)
C
230     F(I,MP1)=F(I,MP1)-AFACT*F(JJ,M)
C
C****   WHEN M=N-2, ELIMINATE ELEMENTS IN BLOCK D(N)
C
        IF (M .NE. NM2) GO TO 270
        DO 260 I=1,K
        DFACT=D(I,JJ,MP2)/B(JJ,JJ,M)
C
C--     COMPUTE NEW VALUES IN D(M+2)
C
        DO 240 J=JJ,K
        D(I,J,MP2)=D(I,J,MP2)-DFACT*B(JJ,J,M)
240     CONTINUE
C
C--     COMPUTE NEW VALUES IN A(M+2), B(M+2), AND F(M+2)
C
```

```
      DO 250 J=1,K
      A(I,J,MP2)=A(I,J,MP2)-DFACT*C(JJ,J,M)
      B(I,J,MP2)=B(I,J,MP2)-DFACT*D(JJ,J,M)
250   CONTINUE
260   F(I,MP2)=F(I,MP2)-DFACT*F(JJ,M)
270   CONTINUE
C
C****  BACK SUBSTITUTE TO GET FINAL SOLUTION IN F(M), M=1,...,N
C
      DO 330 MM=1,N
      M=N-MM+1
      MP1=M+1
      MP2=M+2
C
      DO 320 II=1,K
      I=K-II+1
      IP1=I+1
C
C--   COMPUTE SUM OF PRODUCTS WHEN MULTIPLYING B(M) BY F(M)
C
      BSUM=0.0
      IF (I .EQ. K) GO TO 290
      DO 280 J=IP1,K
280   BSUM=BSUM+B(I,J,M)*F(J,M)
290   CONTINUE
      CSUM=0.0
      IF (M .EQ. N) GO TO 310
C
C--   COMPUTE SUM OF PRODUCTS WHEN MULYIPLYING C(M) BY F(M+1)
C
      CSUM=0.0
      IF (M .EQ. N) GO TO 310
      DO 300 J=1,K
300   CSUM=CSUM+C(I,J,M)*F(J,MP1)
C
C--   COMPUTE SUM OF PRODUCTS WHEN MULTIPLYING D(M) BY F(M+2)
C
      DSUM=0.0
      IF (M .GE. NM1) GO TO 304
      DO 302 J=1,K
302   DSUM=DSUM+D(I,J,M)*F(J,MP2)
304   CONTINUE
310   SUM=BSUM+CSUM+DSUM
C
C--   COMPUTE AND STORE THE FINAL SOLUTION IN F(M)
C
320   F(I,M)=(F(I,M)-SUM)/B(I,I,M)
330   CONTINUE
      GO TO 360
C
340   WRITE (4,350) JJ,M,BMX,AMX,ZRO
```

```
350     FORMAT (10X,' PIVOT ELEMENT IS . LE . ZRO ', 2I5,3E14.7)
        STOP
C
360     RETURN
C
        END
```

```
C-------------------------------------------------------
C
C RUNGE-KUTTA INTEGRATION SCHEME
C
C PURPOSE
C
C   THE PURPOSE OF THE PROGRAM IS TO INTEGRATION THE
C   SOURCE TERM, DT/DC, USING A FOURTH ORDER RUNGE-KUTTA
C   SCHEME. THE RESULTS OF THE INTEGRATION ARE TABULATED
C   AND STORED IN AN OUTPUT FILE.
C
C
C DESCRIPTION OF PARAMETERS
C
C  CINIT-   INITIAL VALUE OF C-SET TO WHERE FUNCTION IS A
C           MINIMUM (0.933).
C CFINAL-   FINAL VALUE OF C FOR THE FORWARD INTEGRATION
C  CFIN2-   FINAL VALUE OF C FOR THE BACKWARD INTEGRATION
C     CT-   AN ARRAY OF SIZE NJ CONTAINING THE RESULTS
C           OF THE INTEGRATION
C   DTDC-   THE FUNCTION TO BE INTEGRATED
C      H-   THE STEP SIZE OF THE INTEGRATION SCHEME-
C           MUST BE LESS THAN CHANGES IN C.
C ICOUNT-   A COUNTER TO DETERMINE WHEN VALUES SHOULD BE
C           WRITTEN
C   INIT-   THE LOCATION IN THE ARRAY AT WHICH INTEGRATION
C           BEGINS
C     NJ-   NUMBER OF DATA POINTS STORED IN TABLE
C    N2J-   DEPENDS ON NJ-USED TO DETERMINE STEP SIZE
C           OF SAMPLING
C NUMBER-      DETERMINES THE FREQUENCY WITH WHICH VALUES ARE

C               WRITTEN
C   STEP-      THE RATE OF SAMPLING BASED ON THE STEP SIZE H
C              AND NJ
C
C-------------------------------------------------------
C
      PARAMETER NJ=502
      DIMENSION CT(NJ)
C
C
      DTDC(C)=EXP(100./(1. + 6. * C) - 57.) * EXP(39.1)/
     & (C * (1. - C))
      DATA CINIT,H,CFINAL,CFIN2/.933,.0005,.999,.001/
C
C
      OPEN(UNIT=2,NAME='INTE.DAT',FORM='FORMATTED',
     & TYPE='NEW')
      CALL ERRSET(72,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
      N2J=NJ-2
      STEP=1/FLOAT(N2J)
```

```
      INIT=(CINIT + 1. - CFINAL) * FLOAT(N2J) + 1
C
C
      C=CINIT
      CT(INIT)=0.
      ICOUNT=0
      NUMBER=STEP/H
C
C
      I=INIT
C
C FORWARD INTEGRATION BEGINNING AT C= .933
C
10    IF(C .GT. CFINAL) GOTO 999
      ICOUNT=ICOUNT + 1
      AK1=H * DTDC(C)
      AK2=H * DTDC(C + H/2.)
      AK3=AK2
      AK4=H * DTDC(C + H)
      TO=TO + (AK1 + 2. * AK2 + 2. * AK3 + AK4)/6.
      IF (ICOUNT .NE. NUMBER) GOTO 6
      I=I + 1
      CT(I)=TO
      ICOUNT=0
6     C=C + H
      GOTO 10
C
C BACKWARD INTEGRATION BEGINNING WITH C= .933
C
999   TO=0.
      C=CINIT
      ICOUNT=0
      I=INIT
21    IF (C .LT. CFIN2) GOTO 99
      ICOUNT=ICOUNT + 1
      AK1=H * DTDC(C)
      AK2=H * DTDC(C + H/2.)
      AK3=AK2
      AK4=H * DTDC(C + H)
      TO=TO - (AK1 + 2. * AK2 + 2. * AK3 + AK4)/6.
      IF (ICOUNT .NE. NUMBER) GOTO 7
      I=I - 1
      CT(I)=TO
      ICOUNT=0
7     C=C - H
      GOTO 21
99    CT(1)=CT(2)
      CT(NJ)=ABS(CT(1))
      WRITE(2,*)(CT(I),I=1,NJ)
      CLOSE(UNIT=2)
      STOP
      END
```