



# Physikalisch-Technische Bundesanstalt

## DECKBLATT

EU 036.24	Projekt	PSP-Element	Obj. Kenn.	Aufgabe	UA	Lfd. Nr.	Rev.
	N A A N	N N N N N N N N N N	N N N N N N	X A A X X	A A	N N N N	N N
	9K	324.34		EG	RB	0003	00

<b>Titel der Unterlage:</b> STLINE Manual, Version 4.80 632A-03G-01 in der überarbeiteten Fassung vom März 1986 "The Intera Particle Tracking Streamline Model"	Seite:
	I.
	Stand:
	3.86

<b>Ersteller:</b> Intera	Textnummer:
-----------------------------	-------------

Stempelfeld:

PSP-Element TP...9K/21285	zu Plan-Kapitel: 3.9
---------------------------	----------------------

	PL	PL
	24.3.86	24.3.86
	Freigabe für Behörden	

Diese Unterlage unterliegt samt Inhalt dem Schutz des Urheberrechts sowie der Pflicht zur vertraulichen Behandlung auch bei Beförderung und Vernichtung und darf vom Empfänger nur auftragsbezogen genutzt, vervielfältigt und Dritten zugänglich gemacht werden. Eine andere Verwendung und Weitergabe bedarf der ausdrücklichen Zustimmung der PTB.

# Revisionsblatt



EU 036.24	Projekt	PSP-Element	Obj. Kenn.	Aufgabe	UA	Lfd. Nr.	Rev.
	N A A N	N N N N N N N N N N	N N N N N N	X A A X X	A A	N N N N	N N
	9K	324.34		EG	RB	0003	00

<b>Titel der Unterlage:</b> STLINE Manual, Version 4.80 632A-03G-01 in der überarbeiteten Fassung vom März 1986 "The Intera Particle Tracking Streamline Model"	<b>Seite:</b> II.
	<b>Stand:</b> 3.86

Rev.	Revisionsst. Datum	verant. Stelle	Gegenzeichn. Name	rev. Seite	Kat. *)	Erläuterung der Revision

\*) Kategorie R = redaktionelle Korrektur  
 Kategorie V = verdeutlichende Verbesserung  
 Kategorie S = substantielle Änderung  
 Mindestens bei der Kategorie S müssen Erläuterungen angegeben werden.

USER'S MANUAL FOR STLINE, VERSION 4.80

632A-03G-01

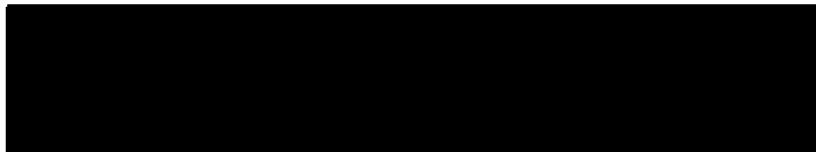
(revised on March 1986)

THE INTERA PARTICLE TRACKING  
STREAMLINE MODEL

Prepared for

TECHNISCHE UNIVERSITÄT BERLIN

By



April, 1980

## TABLE OF CONTENTS

	<u>Page</u>
I      PROGRAM DESCRIPTION	1
II     INPUT DATA GUIDE	5

## I. PROGRAM DESCRIPTION

The streamline model provides an efficient and economical means of computing particle-movement trajectories, distances and times within a steady-state velocity field. After establishing initial particle locations the program "tracks" the movement of individual particles through time. The trajectory (or streamline) of each particle is stored in memory and displayed as a trace of discrete points as viewed in each of the three orthogonal planes.

The program, in present form, is written in a flexible input format. Grid block dimensions, velocities and porosities are prescribed from either formatted cards or unformatted tape, the latter being generated from the execution of the SWIFT program.

### NUMERICAL IMPLEMENTATION

The translation of a particle may be described simply as

$$x = v t$$

that is the distance traveled ( $x$ ) equals the product of the velocity ( $v$ ) and the time increment ( $t$ ). As the velocity may vary spatially, further explanation is necessary.

Implementation of a particle tracking concept is most easily explained in one dimension. Because of orthogonality, extension to higher dimensions is straightforward. The velocity field, computed from potential differences in a finite-difference scheme, such as SWIFT, is defined at grid block edges. Within a grid-block the velocity is obtained by linear interpolation:

$$v = v_1 + (v_2 - v_1) \frac{x}{\Delta x} \quad (1)$$

where  $v_1$  and  $v_2$  are the velocities at the grid-block edges separated by a distance  $\Delta x$ ,  $v$  is the average velocity at the location,  $x$ , the local grid-block coordinate (Figure 1).

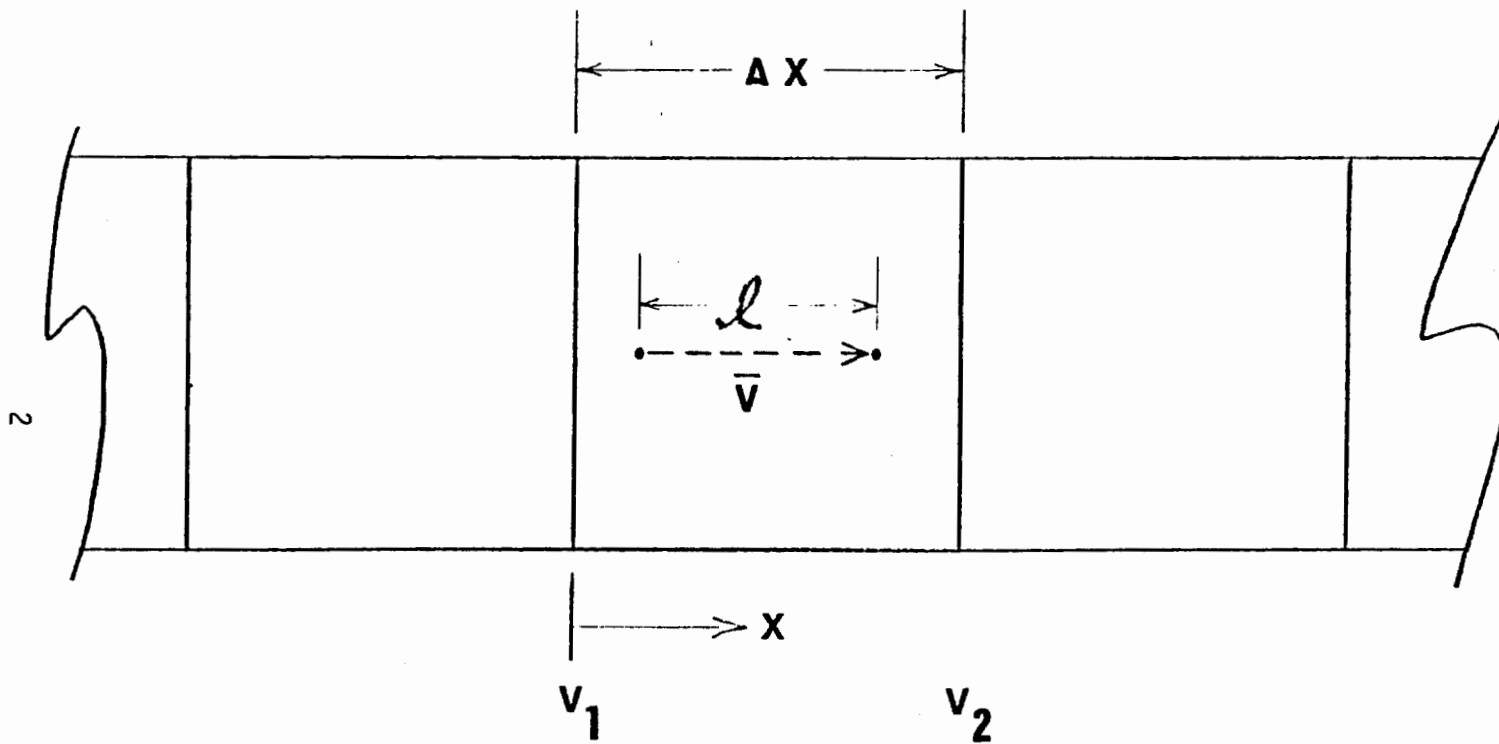


Figure 1. Translation of a particle at velocity  $v$  over a distance  $l$  within a grid-block having edge velocities  $v_1$  and  $v_2$ .

Thus one can define the distance traveled within a time step as

$$l = \bar{v} \cdot \Delta t \quad (2)$$

where  $\Delta t$  is the chosen time increment to move the distance  $l$  at the average velocity,  $\bar{v}$ . Since the velocity is assumed to vary linearly over the grid block, the average velocity over the time increment is the arithmetic average of the two velocities at  $(x)$  and  $(x+l)$ .

$$l = \left[ v_1 + (v_2 - v_1) \frac{x}{\Delta x} + v_1 + (v_2 - v_1) \frac{(l+x)}{\Delta x} \right] \frac{1}{2} \Delta t \quad (3)$$

By rearrangement of Eqn. (3), one can solve for the distance traveled as

$$l = \left\{ \left[ v_1 + (v_2 - v_1) \frac{x}{\Delta x} \right] \Delta t / \left[ 1 - (v_2 - v_1) \Delta t / 2 \Delta x \right] \right\} \quad (4)$$

If during the time step the particle moves out of the grid block, Eq. (4) simply extrapolates velocities linearly.

The program accumulates a path length for each particle. Furthermore, new particle locations are computed and the elapsed simulation time is incremented. The "time-marching" procedure continues until one of the following conditions is met :

- (1) the maximum number of allowable time steps is exceeded,
- (2) all of the particles have exited from the flow field,
- (3) the tolerance on particle movement is exceeded.

The tolerance on particle movement is defined as the fractional degree of movement during a given time step along one of the principal directions. More explicitly, this is defined as the maximum change in position ordinate divided by the total length of the system in any one of the principal directions. This criteria is useful in reducing and eliminating computations in a convergent flow field.

An option has been provided in the establishment of boundary velocities. As these typically are not explicitly defined along outer grid-block edges from potential flow computations by SWIFT, this option allows the establishment of outer edge

velocities equal to the adjacent interior velocity. The choice of velocity specification should be consistent with boundary conditions used to compute the interior velocity field.

~~Point initialization (initial particle location)~~ is defined by a real number. The whole number is that of the grid block and the remaining fraction being the fractional distance across that grid block. The centroid of the first grid block (1,1,1) would be defined with initial ordinates [1.5, 1.5, 1.5].



## II. DATA INPUT GUIDE

### Read R1 (20A4)

LIST: ~~TITLE~~

TITLE        One card of alphanumeric data to serve as a  
title for this run.

### Read R2 (5I5)

LIST: NX, NY, NZ, NP, NTM

NX            Number of grid blocks in x-direction  
NY            Number of grid blocks in y-direction  
NZ            Number of grid blocks in z-direction  
NP            Number of particles be to tracked  
NTM           Maximum number of displayed points per particle

### Read R3 (4I5)

LIST: IRDX, IRVX, IRPHI, IDBUG

IRDX         Control for input of grid block dimensions  
          0        Read from cards  
          1        Read from Tape 3

IRVX         Control for input of Darcy velocities  
          0        Read from cards  
          1        Read from Tape 3  
         -1        Read from Tape 3 and  
                  change direction

IRPHI        Control parameter for input of porosities  
         -1        Uniform porosity read from cards  
          0        Variable porosity read from cards  
          1        Variable pore volume read from Tape 3

IDBUG        Control parameter additional diagnostic printout  
          0        No print  
          1        Print

Read R4

NOTE: Skip this read if IRDX equals one.

LIST: ~~NXX, NYY, NZZ, DX(NXX), DY(NYY), DZ(NZZ)~~

NXX Number of grid blocks in x-direction  
NYY Number of grid blocks in y-direction  
NZZ Number of grid blocks in z-direction  
DX Grid block dimensions in x-direction  
DY Grid block dimensions in y-direction  
DZ Grid block dimensions in z-direction

Read R5

NOTE: Skip this read if IRPHI equals one.

LIST: PHI(NB)

PHI Pore volume of each grid block where NB =  
(NX)(NY)(NZ) = number of grid blocks.

Read R6

NOTE: Skip this read if IRVX equals one.

LIST: UX(NB, UY(NB), UZ(NB)

UX Darcy velocity at leading block edge in x-direction  
UY Darcy velocity at leading block edge in y-direction  
UZ Darcy velocity at leading block edge in z-direction

Read R7 (8E10.0)

NOTE: Skip this read if IRDX equals one.

LIST: DX(NX), DY(NY), DZ(NZ)

DX Grid block dimensions in the x-direction  
DY Grid block dimensions in the y-direction  
DZ Grid block dimensions in the z-direction

(Each array begins on a new card)

Read R8 (E10.0)

NOTE: Skip this read if IRPHI does not equal -1.

LIST: PHII

PHII Uniform value of porosity

Read R9 (8E10.0)

NOTE: Skip this read if IRPHI does not equal 0.

LIST: PHI(NB)

PHI Porosity

Read R10 (8E10.0)

NOTE: Skip this read if IRVX equals one.

LIST: UX(NB), UY(NB), UZ(NB)

UX Darcy velocity at leading block edge in x-direction

UY Darcy velocity at leading block edge in y-direction

UZ Darcy velocity at leading block edge in z-direction

(Each array begins on a new card)

READ R11 (3E10.0)

LIST: XI(NP), YI(NP); ZI(NP)

XI Initial particle location, x-ordinate

YI Initial particle location, y-ordinate

ZI Initial particle location, z-ordinate

[1.5, 1.5, 1.5] is centroid of block (1,1,1)

[1.0, 1.0, 1.0] is the outside corner of block (1,1,1)

A total of NP cards are required.

Read R12 (2E10.0, 110)

LIST: DELT, TOL, NTRED

DELT Time increment for display of streamlines  
TOL Tolerance on particle movement  
(Default = 0.005)  
NTRED Number of internal particle translations  
per display time ( $\Delta t = \text{DELT}/\text{NTRED}$ )

Read R13 (6I5)

LIST: IBCX1, IBCX2, IBCY1, IBCY2, IBCZ1, IBCZ2

IBCX1 Control parameter to establish the boundary  
velocity at  $I = 1$   
IBCX2 Control parameter to establish the boundary  
velocity at  $I = NX$   
0 Boundary velocities in x-direction equals zero.  
1 Adjacent block velocities in x-direction are used  
at boundary edge.

Similar definitions for IBCY and IBCZ.

Read R14 (6I5)

LIST: IDX, IVX, IVY, IVZ, IPHI, IPNT

Print control parameters

0 No print  
1 Print

IDX Grid block dimensions  
IVX Velocity in x-direction  
IVY Velocity in y-direction  
IVZ Velocity in z-direction  
IPHI Porosity  
IPNT Particle locations

Read R15 (2I5)

LIST: LCI, ILN

LCI Number of lines (in multiples of ten) on  
printer plotter in y- or z-direction down  
the page for a x-y and x-z view, respectively.

ILN Control parameter for printing of grid block  
boundary location on streamline plot

0 Do not print boundary locations

1 Print boundary locations

Read R16 (6E10.0)

LIST: XO, XM, YO, YM, ZO, ZM

XO Minimum ordinate in x-direction

XM Maximum ordinate in x-direction

(Similar definitions for YO, YM, ZO, ZM)

Read R17 (40A1)

LIST: XAX, YAX, ZAX

XAX x-axis label up to 40 characters

(Similar definitions for YAX, ZAX with one  
title per card)

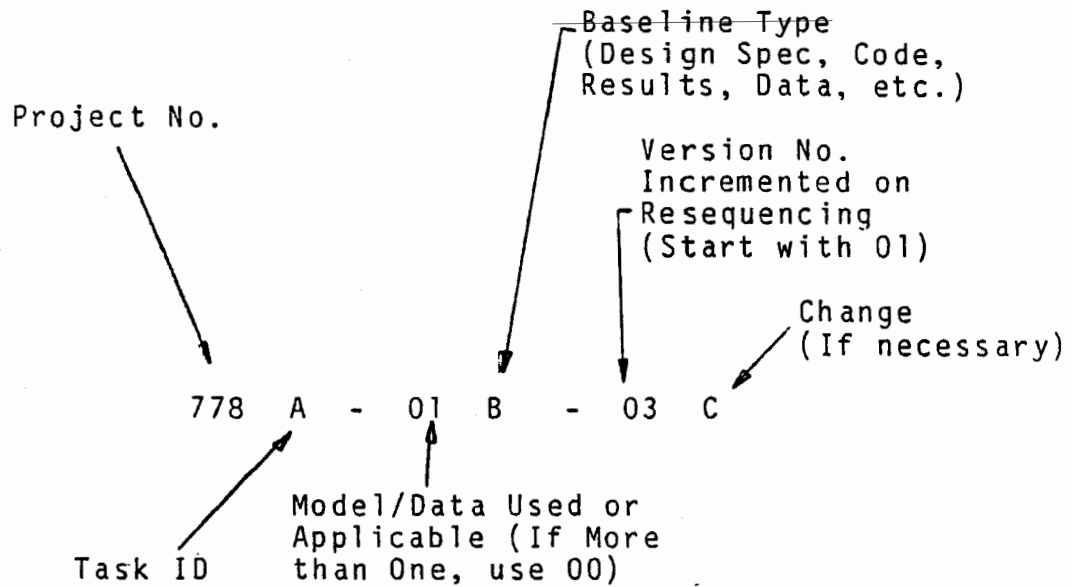
## SUMMARY OF QUALITY ASSURANCE (QA) CONTROL IDENTIFICATION NUMBERING SYSTEM

This section serves to aid the client in the use and understanding of the Quality Assurance (QA) numbering system as outlined in the INTERA QA Program and detailed in the INTERA QA Manual. Through its QA program, INTERA assures that the products and services which it delivers are of sound technical and scientific quality, satisfy contract requirements, are sufficiently well standardized and documented to assure their utility in the hands of INTERA's clients, and supported by a record of their evolution which demonstrates that quality. The Program Plan and Manual are intended for internal use, although they are available for client review where required.

Control of baseline documentation and documentation of selected QA activities are accomplished through a series of QA control documents, or forms. The forms serve as cover pages for the various document sets, e.g., specifications, and provide the formats for documenting QA activities, e.g., problem reports and reviews. These control documents identify the documents or activities involved, identify the Project and component tasks, establish control ID numbers (ID #) applicable to the documentation or activity involved, and record the required recommendations, approvals and other pertinent information.

Each control document relies on its title and a control ID # for identification. The ID # is constructed of a project identifier which corresponds to INTERA's accounting numbering system, a sequential task identifier, and sequential model, task phase, and version identifiers in the following format:

## Control Identification Numbering System



The project identifier is composed of three digits which are identical with those used in INTERA's accounting system. The task identifier is the alpha subscript. The model identifier is a two-digit number, and an alpha subscript indicates the type of baseline according to the following system:

- A - Performance Specifications
- B - Design Specifications
- C - Code
- D - Test Specifications
- E - Test/Application Data
- F - Test/Application Results
- G - Reports
- H - Problems
- I - Reviews
- J - Audits

The version identifier is a two-digit number; the version identifier subscript is used only to identify baseline version change documents. When a document is not task-specific

or baseline-type specific, the corresponding alpha subscript is omitted. When the document is not model-specific, the model identifier is 00.

Each detached page of documentation must contain the control ID # in the upper right hand corner.

The system of control documents is administered by the QA-AA, who maintains the Control Library. The Control Library contains the originals of all controlled documentation and all control documents. An example inventory statement for the Control Library is attached to this Manual. The QA-AA assures that all control documents are properly identified and completed. The QA-AA and Intera Accounting assign the project number; other portions of the ID # flow from the PQAP and numerical or alphabetical sequence. The QA-AA maintains a distribution list for all controlled documents and control documents.