KfK 4692
Februar 1990

# Parallel Treatment of Simulation Particles in Particle-in-Cell Codes on SUPRENUM

D. Seldner
Institut für Datenverarbeitung in der Technik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Datenverarbeitung in der Technik

KfK 4692

# Parallel Treatment of Simulation Particles in Particle-in-Cell Codes on SUPRENUM *

David Seldner

* Das diesem Bericht zugrundeliegende Vorhaben wurde aus Mitteln des Bundesministers für Forschung und Technologie unter dem Förderkennzeichen ITR8502K/4 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

## Parallele Behandlung von Simulationsteilchen in Particle-in-Cell Codes auf SUPRENUM


### Zusammenfassung

Vorliegender Bericht enthält die Programmdokumentation und -beschreibung des Arbeitspakets 2D-PLAS, das im Rahmen des vom BMFT geförderten SUPRENUM-Projekts im Kernforschungszentrum Karlsruhe im Institut für Datenverarbeitung in der Technik (IDT) erstellt wurde. 2D-PLAS ist eine parallele Programm-Version der Behandlung der Simulationsteilchen des im KfK entwickelten zweidimensionalen stationären Particle-in-Cell Codes BFCPIC, die speziell für den Parallelrechner SUPRENUM entwickelt wurde.


## Parallel Treatment of Simulation Particles in Particle-in-Cell Codes on SUPRENUM


### Summary

This report contains the program documentation and description of the program package 2D-PLAS, which has been developed at the Nuclear Research Center Karlsruhe in the Institute for Data Processing in Technology (IDT) under the auspices of the BMFT. 2D-PLAS is a parallel program version of the treatment of the simulation particles of the two-dimensional stationary particle-in-cell code BFCPIC which has been developed at the Nuclear Research Center Karlsruhe. This parallel version has been designed for the parallel computer SUPRENUM.

## Contents:

# 1. Overview

## 1.1 The Particle-in-Cell Code

In order to simulate pulsed power ion diodes the two-dimensional stationary particle-in-cell code based on boundary-fitted coordinates, BFCPIC, has been developed at the Nuclear Research Center Karlsruhe. The code serves to support the experimental investigation of the diodes. By means of this PIC code pulsed power ion diodes are modelled in order to obtain a better understanding of the phenomena. There exists a two-dimensional stationary version [3,4]. For parameter studies with respect to the geometry, many simulations with varying anode-cathode-gap distances, applied voltages, etc. must be performed.

As the geometries considered have rotational symmetry, cylindrical coordinates (r, $\phi$, z) are introduced in order to describe the geometry. As a result of symmetry considerations, no $\phi$-dependence is obtained for self-magnetically insulated ion-diodes. Thus, for these diodes a two-dimensional model is sufficient. The geometry of the diode is described by means of boundary-fitted coordinates [5]. In spite of the reduction to two dimensions, the simulations require extremely large computer times because of the large number of particles involved ($10^4$ - $10^5$), even when a very coarse grid (less than 250 grid points) is used.

Essentially the following variables are needed for such a particle-in-cell code:

Grid quantitites: electric fields in z- and r- direction, magnetic field in $\Phi$-direction:

$E_z, E_r, B_\Phi$ ,

stored as two-dimensional arrays E1, E2, B3 .

Particle quantities: coordinates and velocity in z- and r-direction, charge, location in the grid (cell address plus weight):

$z, r, v_z, v_r, q, I + \alpha_1, J + \alpha_2$ ,

stored in the particle data matrices PS consisting of 7 columns.

The positions of the particles as well as the fields are given at full time levels, i.e. at points of time $n \cdot \Delta t$, if $\Delta t$ denotes the length of the time step and $n = 0, 1, 2, \ldots$ is the time level. The velocities, however, are given half a time step earlier (at points of time $(n - \frac{1}{2}) \cdot \Delta t$ ) due to the used leapfrog algorithm for the particle pushing. The velocities are extrapolated half a time step in order to compute charge and current densities at full time levels.

The structure of the code is standard [1,2] and the individual modules are briefly described below. For an algorithmic description we refer to [6-12]:

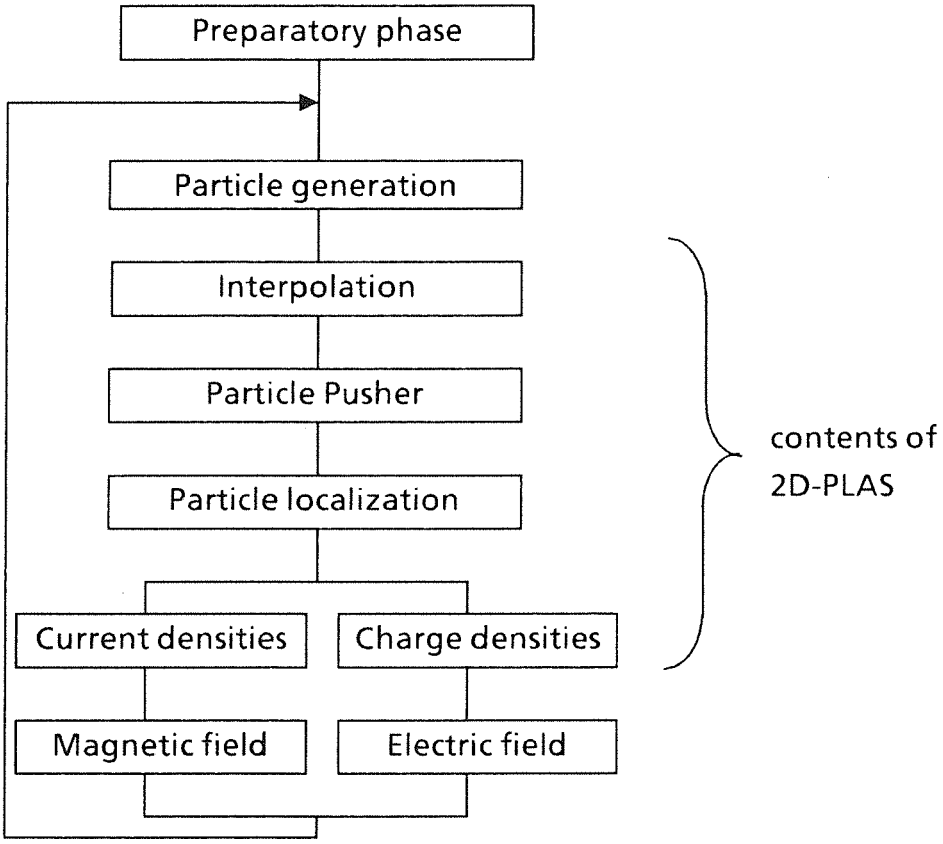


Fig. 1: Principle scheme of the particle-in-cell code.

*Preparatory phase*: This initial step consists in identifying electrode cells, determining the volumes of the cells, computing the fields in the empty diode, and so on.
*Particle generation*: Due to the electric field existing at the electrodes new particles (electrons, ions) are emitted there.
*Interpolation*: To be able to move the particles the fields must be known at the particle positions. Therefore, the fields determined at the nodes of the grid are interpolated onto the particle positions.
*Particle pusher*: Every particle is moved by means of the Lorentz forces acting upon it. Depending on the type of the particles (electrons, ions) the movement is either relativistic or non-relativistic.
*Localization*: In order to obtain the data necessary for the field calculations every particle must be localized inside the grid and interpolation weights be assigned to the particles.
*Current and charge densities*: With the weights determined in the localization step the current and charge densities at the grid points are calculated.
*Fields*: By means of the current density at the grid points the magnetic field is computed with the Ampère law. The electric potential is computed by solving the Poisson equation using multigrid methods. The electric field is determined from the

potential by numerical differentiation (for this computation program package 2D-DIO can be applied).

The most time-consuming part of the code, the treatment of the simulation particles (i.e., the modules interpolation, particle pusher, localization, and computation of charge and current densities), has been parallelized for SUPRENUM (cf. [14,15,16]) and is contents of the program package 2D-PLAS.

Compatibility to the existing scalar BFCPIC code [3] is a main feature of 2D-PLAS. The treatment of the particles is carried out using the same structure and subroutines as in the original code. Communication and administration concerning the parallelization are performed in subroutines described in chapter 2. Sending and receiving of the data to and from the particle processes as well as the control of the particle processes can be performed either by the host process or by a process running on a node. See chapter 4 for a description of the algorithms and chapter 5 for programming details.

## 1.2 Purpose of the program

The purpose of the task program PUSH is mainly to advance electrically charged macro-particles (electrons, ions) in electric and magnetic fields and to compute charge and current densities resulting from the phase-space coordinates of the particles.

The force acting upon the particles is determined by an interpolation from the fields at the nodes of the grid (the main input). Vice versa, to obtain the densities in the grid points by means of the charge of the particles, the particles are localized within the grid.

The cells of the corresponding logical grid consisting of I1MX points in 1-direction and I2MX points in 2-direction must be numbered as shown in figure 2.

Each particle stores the address of the cell it is located in together with interpolation weights $(\alpha_1,\alpha_2) \in I^2$ (the unit square) determined as described in chapter 4.

Main input for the program package 2D-PLAS are in the beginning the particle data and thereafter the electric and magnetic fields on the nodes of a (possibly irregular) grid and - if desired - new particles. If new particles are sent to the particle processes, "holes" in the particle data matrices are filled with these particles (i.e., particles that have been outside the diode for more than one time step are replaced; if there are not enough "holes" in the matrices, the particles are added).

Main output are the charge and current densities of the electrons and ions and - if desired - the particle data matrices.

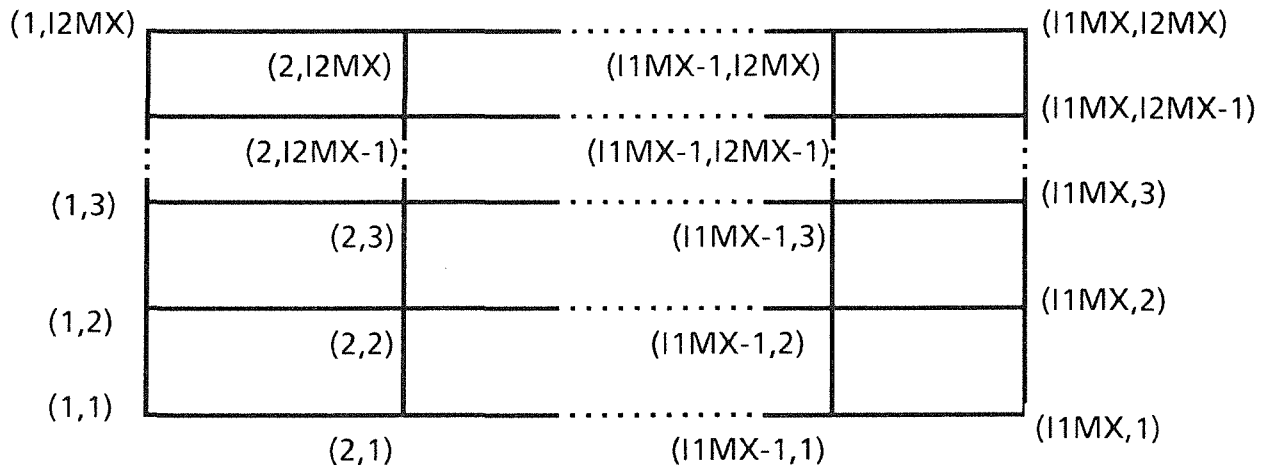


Figure 2: Addresses (numbering) of the cells

The particle data must be stored in matrices PS*i*, where *i* stands for the type of particles: $i = 1$: electrons, $i = 2,...,k \leqq 6$: ions of species k. The particle data matrices are two-dimensional arrays dimensioned as PS*i*(NPS*i*MX,7) with *i* as above. The meaning of the 7 columns are described in the following for the example of the electrons:

PS1(I,1) : coordinate of particle I in 1-direction
PS1(I,2) : coordinate of particle I in 2-direction
PS1(I,3) : velocity of particle I in 1-direction
PS1(I,4) : velocity of particle I in 2-direction
PS1(I,5) : charge of particle I
PS1(I,6) : cell address and weight of particle I in 1-direction
INT(PS1(I,6)) indicates the cell address in 1-direction
PS1(I,6) - INT(PS1(I,6)) indicates the interpolation weight in 1-direction
PS1(I,6) = -1.1 : particle I has been detected outside the grid during the last time step
PS1(I,6) = -2.1 : particle I has been outside the grid for more than one time step
PS1(I,7) : cell address and weight of particle I in 2-direction
INT(PS1(I,7)) indicates the cell address in 2-direction
PS1(I,7) - INT(PS1(I,7)) indicates the interpolation weight in 2-direction
PS1(I,7) = -1.1 : particle I has been detected in a cell with volume 0
PS1(I,7) = -2.1 : particle I is on the right hand side of the right boundary
PS1(I,7) = -3.1 : particle I is over the upper boundary
PS1(I,7) = -4.1 : particle I is on the left hand side of the left boundary
PS1(I,7) = -5.1 : particle I is below the lower boundary

# 2. Usage

In this section first the structure of the program is introduced. After a summary of the commonly used variables of the BFCPIC code, the usage of the individual subroutines is described.

## 2.1 Structure of the program

The processes designed to treat particles, the so-called particle processes (task program name PUSH) are created and assigned with data by the host process by calling subroutine CRENOD. At the beginning of a time step, the new particles and the electric and magnetic fields are sent to the particle processes (subroutine SNEW). Each particle process treats its particles as described in the previous chapter and finally computes (partial) charge and current densities. These partial densities are sent to the host together with other data; the host receives this data by calling subroutine RDENS. In RDENS the partial densities are summed up. With the supplied data the field calculations can be carried out. In PTCLSR the number of particles and other diagnostic values can be received. By calling subroutine MINMAN the load balance of the particle processes can be checked and, if necessary, an exchange of particles between the particle processes is initiated. Subroutine RPTCLS provides the possibility to receive the particle data matrices. If many particles are treated it is advisible not to call this subroutine very often. The sequence of the subroutine calls and the structure is outlined in fig. 3 and 4.

By using the available controlling process, task program name STEUER, the time step loop can be carried out on a node. STEUER supplies the particle processes with data, receives the results, performs diagnostic computations, and sends these results to the host. See the documentation of STEUER for details.

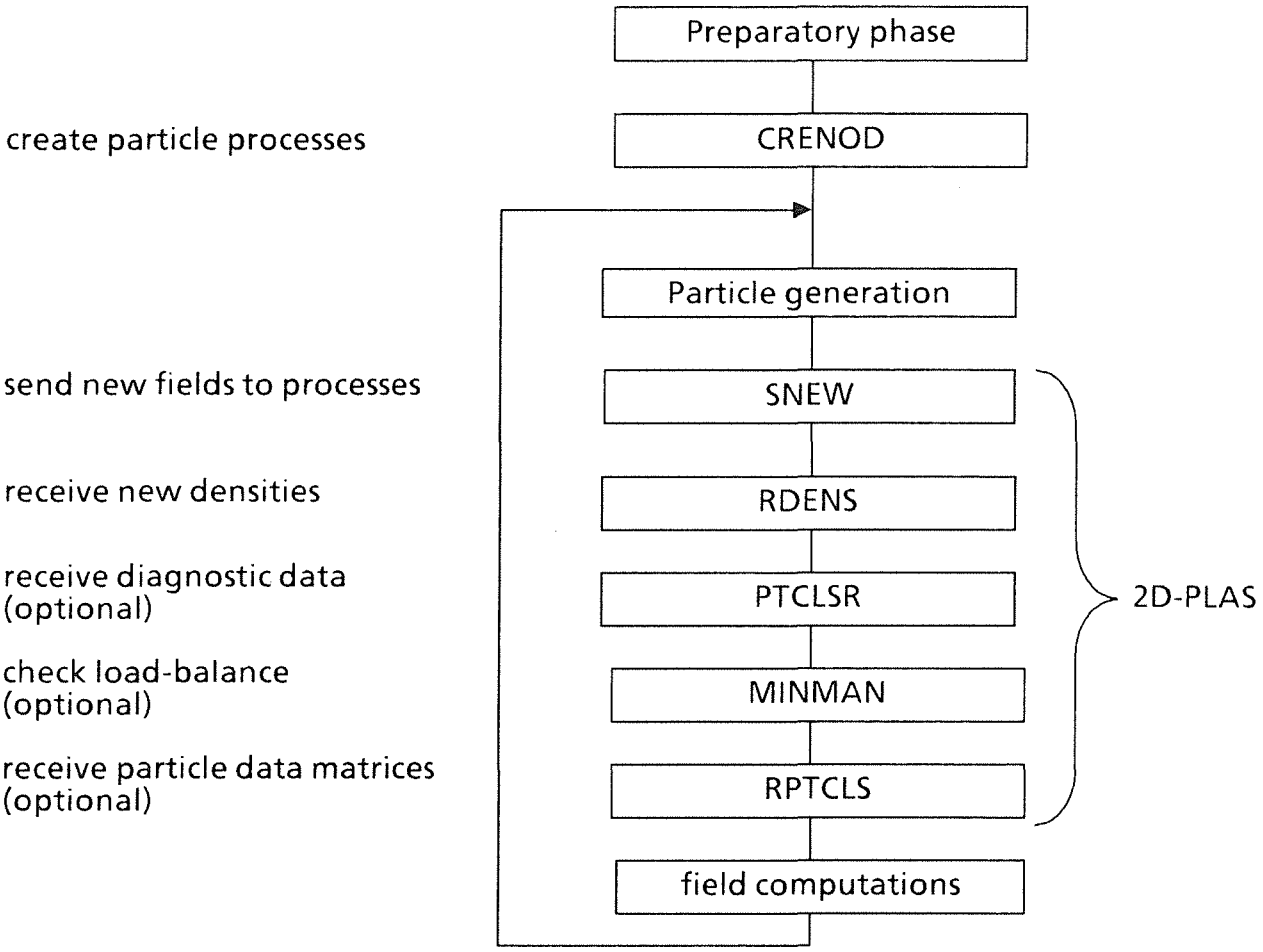


Fig. 3: Scheme of a particle-in-cell code with parallelized treatment of simulation particles.

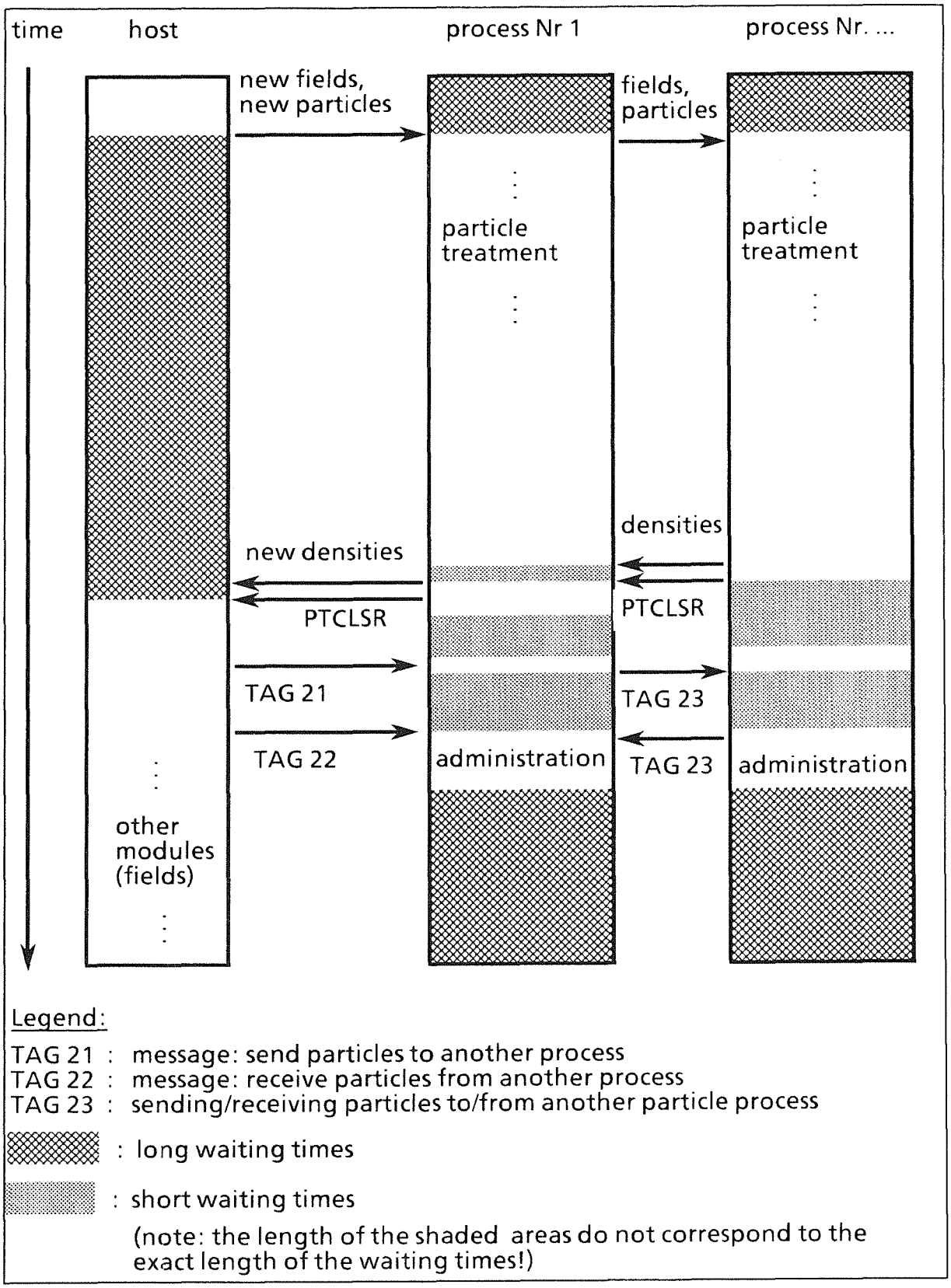


Fig. 4: time history of communication.

## 2.2 Commonly used variables

| Name | function | type | dimension |
|---|---|---|---|
| **NPS*i*MX**<br>(*i* = 1,...,6) | ***Input***<br>maximum number of particles of species *i* (dimension of field PS*i*) | **INTEGER** | |
| **NPS*i***<br>(*i* = 1,...,6) | ***Input/Output***<br>actual number of particles of species *i* in the diode | **INTEGER** | |
| **PS*i***<br>(*i* = 1,...,6) | ***Input/Output***<br>particle data matrix (species *i*)<br>see also page 4 | **REAL** | **(NPS*i*MX,7)** |
| **ANZHLE** | ***Output***<br>number of electrons in the particle processes (ANZHLE(I), I = 1,...,NPROC , is the number in process PROC(I)) | **INTEGER** | **(MXPROC)** |
| **ANZHL*i***<br>(*i* = 2,...,6) | ***Output***<br>number of ions of species *i* inthe particle processes (ANZHL*i*(I), I = 1,...,NPROC , is the number in process PROC(I)) | **INTEGER** | **(MXPROC)** |
| **NDTE** | ***Input***<br>number of small electron time steps with new localization per large ion time step | **INTEGER** | |
| **NNDTE** | ***Input***<br>number of small electron time steps before new localization | **INTEGER** | |
| **NDIM1** | ***Input***<br>maximum number of grid points in 1-direction (dimension) | **INTEGER** | |
| **NDIM2** | ***Input***<br>maximum number of grid points in 2-direction (dimension) | **INTEGER** | |

| Name | function | type | dimension |
|---|---|---|---|
| I1MX | *Input* | INTEGER | |
| | actual number of grid points in 1-direction (I1MX≦NDIM1) | | |
| I2MX | *Input* | INTEGER | |
| | actual number of grid points in 2-direction (I2MX≦NDIM2) | | |
| X1 | *Input* | REAL | (NDIM1,NDIM2) |
| | coordinates of the grid points in 1-direction | | |
| X2 | *Input* | REAL | (NDIM1,NDIM2) |
| | coordinates of the grid points in 2-direction | | |
| V | *Input* | REAL | (NDIM1,NDIM2) |
| | inverse volumes of the cells | | |
| E1 | *Input* | REAL | (NDIM1,NDIM2) |
| | electric field at the grid points in 1-direction | | |
| E2 | *Input* | REAL | (NDIM1,NDIM2) |
| | electric field at the grid points in 2-direction | | |
| B3 | *Input* | REAL | (NDIM1,NDIM2) |
| | magnetic field at the grid points in 3-direction | | |
| RHOE | *Output* | REAL | (NDIM1,NDIM2) |
| | electron charge density in the diode | | |
| AJ1E | *Output* | REAL | (NDIM1,NDIM2) |
| | electron current density in 1-direction in the diode | | |
| AJ2E | *Output* | REAL | (NDIM1,NDIM2) |
| | electron current density in 2-direction in the diode | | |
| RHOI | *Output* | REAL | (NDIM1,NDIM2) |
| | ion charge density in the diode | | |

| Name | function | type | dimension |
|---|---|---|---|
| AJ1I | *Output* | REAL | (NDIM1,NDIM2) |
| | ion current density in 1-direction in the diode | | |
| AJ2I | *Output* | REAL | (NDIM1,NDIM2) |
| | ion current density in 2-direction in the diode | | |
| RHOEX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| AJ1EX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| AJ2EX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| RHOIX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| AJ1IX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| AJ2IX | | REAL | (NDIM1,NDIM2) |
| | workspace (storage of densities) | | |
| EDM*i* | *Input* | REAL | |
| (*i* = 1,...,6) | charge-to-mass ratio of the particles of species *i* | | |
| NELKR | *Input* | INTEGER | |
| | number of cells with volume 0 ("forbidden" cells, e.g inner electrode cells) | | |
| DT | *Input* | REAL | |
| | length of time step | | |
| DTE | *Input* | REAL | |
| | length of small electron time steps | | |

| Name | function | type | dimension |
|---|---|---|---|
| **NPSORT** | ***Input*** | **INTEGER** | |
| | number of particle species ($1 \leqq$ NPSORT $\leqq 6$) | | |
| **IERROR** | ***Output*** | **INTEGER** | |
| | error code (see 2.4) | | |
| **MES** | ***Input*** | **INTEGER** | |
| | parameter for output of messages in STOPIT (see description of STOPIT) | | |

## 2.3 Usage of the subroutines

### 2.3.1 SUBROUTINE CRENOD

Function:

Creation of particle processes (task program name PUSH) and sending of data to these processes. This subroutine must be called by the initial process.

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

STOPIT, if there is a message with TAG = 0 in the mailbox (i.e., an error has occurred in a node process)

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL CRENOD (MXPROC, PROC, HAUPT, NPROC, NSEND, IDEFLT,
     -             NPS1MX, NPS2MX, NPS3MX, NPS4MX, NPS5MX, NPS6MX,
     -             NPS1, NPS2, NPS3, NPS4, NPS5, NPS6, PS1, PS2, PS3,
     -             PS4, PS5, PS6, ANZHLE, ANZHL2, ANZHL3, ANZHL4,
     -             ANZHL5, ANZHL6, NDIM1, NDIM2, I1MX, I2MX, X1, X2,
     -             V, EDM1, EDM2, EDM3, EDM4, EDM5, EDM6, NELKR, DT,
     -             DTE, NPSORT,-IERROR, MES)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes (dimension) | | |
| **PROC** | ***InlOutput*** | **TASKID** | **(MXPROC)** |
| | identification of particle processes | | |
| **HAUPT** | ***input*** | **TASKID** | |
| | identification of the controlling process (the process the particle processes have to send their data to) | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | desired number of particle processes (NPROC $\leqq$ MXPROC) | | |
| **NSEND** | ***Output*** | **INTEGER** | **(MXPROC)** |
| | work load of the particle processes (if the sum of the load in the particle processes is not equal 1, this is due to rounding errors) | | |
| **IDEFLT** | ***Input*** | **INTEGER** | |
| | $\geqq 1$: particle processes are created and data is sent | | |
| | $= 0$: only data is sent | | |

### 2.3.2 SUBROUTINE SNEW

Function:

Send of fields and new particles to particle processes. If new particles are to be sent they must be stored in the particle data matrices PS*i*, *i* = 1,...,NPSORT. The data is sent to the particle processes with the identifications PROC(KK), where

$KK = 2^{i-1}$, i = 1,...,NSND, if $KK \leqq$ NPROC

with NSND = INT(ALOG(FLOAT(MAX(1,NPROC)))/ALOG(2.)) + 1.

Some of these particle processes send the data to other particle processes (according to the tree structure described in chapter 5).

Programming language

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL SNEW (MXPROC, PROC, PSSEND, NPROC, NPS1MX,
     -           NPS1, PS1, NPS2MX, NPS2, PS2, NPS3MX, NPS3,
     -           PS3, NPS4MX, NPS4, PS4, NPS5MX, NPS5, PS5,
     -           NPS6MX, NPS6, PS6, NDTE, NNDTE, ENDE, MCOMPR,
     -           NDIM1, NDIM2, I1MX, I2MX, E1, E2, B3, NSEND,
     -           NSND)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes (dimension) | | |
| **PROC** | ***Input*** | **TASKID** | **(MXPROC)** |
| | identification of particle processes | | |
| **PSSEND** | ***Input*** | **LOGICAL** | |
| | if PSSEND = .TRUE. the particle processes will send their particles to the host after the time step | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | number of particle processes | | |
| **ENDE** | ***Input*** | **LOGICAL** | |
| | ENDE = .TRUE.: indicates that this is the last time step; after the time step the particle processes terminate | | |
| **MCOMPR** | ***Input*** | **INTEGER** | |
| | if MCOMPR = 1 the particle matrices are compressed (particles outside the grid are eliminated) | | |
| **NSEND** | ***Input*** | **INTEGER** | **(MXPROC)** |
| | NSEND(I) is the number of nodes of the sub-tree, whose root is PROC(I) (i.e., NSEND(I)-1 indicates to how many processes PROC(I) has to send data) | | |
| **NSND** | ***Input*** | **INTEGER** | |
| | maximum number of sends (levels of the tree) | | |

### 2.3.3 SUBROUTINE RDENS

Function:

Receive of charge and current densities from the particle processes. The data is received from the particle processes with the identifications PROC(KK), where

$KK = 2^{i-1}$, $i = 1,...,NSND$, if $KK \leqq NPROC$

(according to the tree structure of the communication described in chapter 5).

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

**Aufruf:**

```
      CALL RDENS (NPROC, NDIM1, NDIM2, I1MX, I2MX, RHOE, AJ1E,
     -           AJ2E, RHOI, AJ1I, AJ2I, RHOEX, AJ1EX, AJ2EX, RHOIX,
     -           NSND, AJ1IX, AJ2IX, Q1, Q2, QAUS, IERROR)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| NPROC | *Input* | INTEGER | |
| | number of particle processes | | |
| NSND | *Input* | INTEGER | |
| | maximum number of sends (levels of the tree) | | |
| Q1 | *Output* | REAL | |
| | total electron charge in the diode | | |
| Q2 | *Output* | REAL | |
| | total ion charge in the diode | | |
| QAUS | *Output* | REAL | |
| | outgoing electron charge | | |

### 2.3.4 SUBROUTINE MINMAN

Function:

Check of the load-balance of the particle processes and, if necessary, initiation of particle exchange between particle processes. This subroutine must not be called during the last time step!

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL MINMAN (MXPROC, PROC, NPROC, IPFUNC, ANZHLE, ANZHL2,
     -             ANZHL3, ANZHL4, ANZHL5, ANZHL6, NPTCLS, IERROR)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes (dimension) | | |
| **PROC** | ***Input*** | **TASKID** | **(MXPROC)** |
| | identification of particle processes | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | number of particle processes | | |
| **IPFUNC** | ***Output*** | **INTEGER** | **(MXPROC)** |
| | work space (indicates which particle processes have exchanged particles) | | |
| **NPTCLS** | ***Input*** | **INTEGER** | |
| | minimum number of particles to be exchanged (if less than NPTCLS particles would be exchanged no exchange is performed) | | |

### 2.3.5 SUBROUTINE PTCLSR

Function:

Receive of parts of the particle data matrices (all particles I with PS*i*(I,6) = -1.1, *i* = 1,...,NPSORT) from the particle processes.

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL PTCLSR (PROC, MXPROC, PS1, NPS1MX, NPS1A, PS2, NPS2MX,
     -             NPS2A, PS3, NPS3MX, NPS3A, PS4, NPS4MX, NPS4A,
     -             PS5, NPS5MX, NPS5A, PS6, NPS6MX, NPS6A, NPS1,NPS2,
     -             NPS3, NPS4, NPS5, NPS6, NPROC, ANZHLE, ANZHL2,
     -             ANZHL3, ANZHL4, ANZHL5, ANZHL6, LAST, NDTE, NNDTE,
     -             IERROR)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **PROC** | ***Input*** | **TASKID** | **(MXPROC)** |
| | identification of particle processes | | |
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes (dimension) | | |
| **NPS*i*A** | ***Output*** | **INTEGER** | |
| | number of received particles of species *i* | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | number of particle processes | | |
| **LAST** | ***Output*** | **INTEGER** | |
| | relative load of the particle processes<br>(LAST(I) = 3*NDTE*NNDTE*ANZHLE(I) + ANZHL2(I)) | | |

### 2.3.6 SUBROUTINE RPTCLS

Function:

Receive of the particle data matrices from the particle processes. If many particles are treated RPTCLS should not be called very often. This subroutine can only be called by the initial process!

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Usage:

```
      CALL RPTCLS (PROC, MXPROC, PS1, NPS1MX, NPS1A, PS2, NPS2MX,
     -             NPS2A, PS3, NPS3MX, NPS3A, PS4, NPS4MX, NPS4A,
     -             PS5, NPS5MX, NPS5A, PS6, NPS6MX, NPS6A, NPS1,
     -             NPS2, NPS3, NPS4, NPS5, NPS6, NPROC, IERROR,
     -             PSFULL, IFS, IFILEO)
```

List of not previously described arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **PROC** | ***Input*** identification of particle processes | **TASKID** | **(MXPROC)** |
| **MXPROC** | ***Input*** maximum number of particle processes (dimension) | **INTEGER** | |
| **NPS*i*A** | ***Output*** number of received particles of species *i* ( = NPS*i* if PSFULL = .FALSE.) | **INTEGER** | |
| **NPROC** | ***Input*** number of particle processes | **INTEGER** | |
| **PSFULL** | ***Output*** indicates whether all particles were received<br>PSFULL = .TRUE. : the particle matrices were too small to receive all particles<br>PSFULL = .FALSE. : all particles were received | **LOGICAL** | |
| **IFS** | ***Output***<br>IFS≧2 : output of particles into file IFILEO if PSFULL = .TRUE.<br>IFS≧3 : output of particles into file IFILEO | **INTEGER** | |
| **IFILEO** | ***Output***<br>file number for the output of the particles<br>IFILEO = 0 : no output (not possible if IFS≧2) | **INTEGER** | |

### 2.3.7 SUBROUTINE STOPIT

Function:

Receive of an error message (i.e., a message with TAG = 0) from a node process. This subroutine can only be called by the initial process!

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

**Usage:**

```
CALL STOPIT (PROC, NPROC, HAUPT, ERROR, MES, ITERM)
```

List of arguments:

| Name | function | type | dimension |
|---|---|---|---|
| PROC | *Input* | TASKID | (MXPROC) |
| | Identification of particle processes | | |
| NPROC | *Input* | INTEGER | |
| | number of particle processes | | |
| HAUPT | *Input* | TASKID | |
| | Identification of the contolling process | | |
| ERROR | *Output* | INTEGER | |
| | error code (see 2.4) | | |
| MES | *Input* | INTEGER | |
| | parameter for output of messages | | |
| | MES = 1 : output of the message | | |
| | MES = 0 : no output | | |
| ITERM | *Output* | INTEGER | |
| | indicates whether the sending process has terminated | | |
| | ITERM = 1 : the sending process has terminated | | |

## 2.4 List of error codes

| Number | task program (and subroutine), reason | user's action |
|---|---|---|
| 0 | no error | none |
| 1003 | **BFCHST (CRENOD)**<br>NPROC = 0 | none |
| 1113 | **CONTROL PROCESS (SNEW)**<br>NPROC = 0 | |
| 1300 | **PUSH (main program)**<br>MAX(NPS2MX, NPS3MX, NPS4MX, NPS5MX, NPS6MX) < NPS1MX | set NPS1MX ≧ MAX(NPS2MX, NPS3MX, NPS4MX, NPS5MX, NPS6MX) |
| 1301 | **PUSH (main program)**<br>NPS1MX < MXPROC | set NPS1MX ≧ MXPROC |
| 1302 | **PUSH (RDATA)**<br>particles do not fit into the particle data matrices | increase the dimensions NPS*i*MX or compress particle data matrices more often |
| 1303 | **PUSH (RIDENT)**<br>not all particles could be received (not severe, no termination) | none |
| 1304 | **PUSH (RDATA)**<br>I1MX > NDIM1 or I2MX > NDIM2 | increase NDIM1 and/or NDIM2 |
| 1305 | **PUSH (main program)**<br>new particles do not fit into the particle data matrices | increase the dimensions NPS*i*MX or decrease NCOMPR |
| 1306 | **PUSH (RSEND)**<br>too many new particles | increase the dimensions NPS*i*MN |
| 1307 | **PUSH (SIDENT)**<br>not all particles could be sent (not severe, no termination) | none |

| | | |
|---|---|---|
| **1311** | **PUSH (PEINS)**<br>particles do not fit into the particle data matrices | increase the dimensions NPS*i*MX or decrease NCOMPR |
| **1312** | **PUSH (PEINST)**<br>particles do not fit into the particle data matrices | increase the dimensions NPS*i*MX or decrease NCOMPR |
| **3001** | **CONTROL PROCESS (SNEW)**<br>not all new particles were sent (internal error) | none |

## 3. Computational Example

After the input of the grid and particle data the particle processes are created by calling CRENOD. In each time step one new electron is injected to the system. 4 time steps are carried out with two particle species (electrons and protons), NPSORT = 2. The electrons are always advanced 5 times without being localized (NNDTE = 5), this procedure is performed 4 times during one ion time step (NDTE = 4). The particle data matrices are not compressed (NCOMPR = 0) and an update of the load balance is performed even when only one particle has to be exchanged (NPTCLS = 1). The particle data is received every second time step and printed. After the time step loop the total charge density is printed.
We first list the program and then the resulting output.

```
      PROGRAM MAIN
C
C*********************************************************************
C******   HAUPTPROGRAMM ZUM TESTEN DER BEHANDLUNG DER TEILCHEN  *******
C*********************************************************************
C *
C *   AUTOR: D. SELDNER
C *          KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH
C *          INSTITUT FUER DATENVERARBEITUNG IN DER TECHNIK
C *          TEL. 82-5595
C *          STAND: 04.12.1989
C *
C*********************************************************************
C
      PARAMETER (NDIM1=41,NDIM2=65,NDELKR=1)
      PARAMETER (NPS1MX=100,NPS2MX=100,NPS3MX=1,NPS4MX=1)
      PARAMETER (NPS5MX=1,NPS6MX=1,NPS1MN=10,NPS2MN=10)
      PARAMETER (NPS3MN=10,NPS4MN=10,NPS5MN=10,NPS6MN=10)
      PARAMETER (MXPROC=64,LAENGE=1000)
C
      REAL X1(NDIM1,NDIM2),X2(NDIM1,NDIM2),V(NDIM1,NDIM2)
      INTEGER IELKR1(NDELKR),IELKR2(NDELKR)
      REAL RHOE(NDIM1,NDIM2),RHOI(NDIM1,NDIM2)
      REAL AJ1E(NDIM1,NDIM2),AJ2E(NDIM1,NDIM2)
      REAL AJ1I(NDIM1,NDIM2),AJ2I(NDIM1,NDIM2)
      REAL RHOEX(NDIM1,NDIM2),RHOIX(NDIM1,NDIM2)
      REAL AJ1EX(NDIM1,NDIM2),AJ2EX(NDIM1,NDIM2)
      REAL AJ1IX(NDIM1,NDIM2),AJ2IX(NDIM1,NDIM2)
      REAL E1(NDIM1,NDIM2),E2(NDIM1,NDIM2),B3(NDIM1,NDIM2)
C
      REAL PS1(NPS1MX,7),PS2(NPS2MX,7),PS3(NPS3MX,7)
      REAL PS4(NPS4MX,7),PS5(NPS5MX,7),PS6(NPS6MX,7)
      REAL PS1N(NPS1MN,7),PS2N(NPS2MN,7),PS3N(NPS4MN,7)
      REAL PS4N(NPS4MN,7),PS5N(NPS5MN,7),PS6N(NPS6MN,7)
C
```

```
      INTEGER ANZHLE(MXPROC),ANZHL2(MXPROC),ANZHL3(MXPROC)
      INTEGER ANZHL4(MXPROC),ANZHL5(MXPROC),ANZHL6(MXPROC)
      INTEGER LAST(MXPROC),IPFUNC(MXPROC)
      TASKID HAUPT,PROC(MXPROC)
      LOGICAL ENDE,PSFULL,PTSSND,NPSSND

      DATA ANZHLE,ANZHL2,ANZHL3/MXPROC*0,MXPROC*0,MXPROC*0/
      DATA ANZHL4,ANZHL5,ANZHL6/MXPROC*0,MXPROC*0,MXPROC*0/
      DATA ENDE,IERROR/.FALSE.,0/
      DATA I1MX,I2MX  /11,21/

      DATA IELKR1,IELKR2 / NDELKR*0, NDELKR*0 /

      DATA EDM1,EDM2,EDM3,EDM4/-1.7588E11,9.579E7,9.579E7,9.579E7/
      DATA EDM5,EDM6/9.579E7,9.579E7/

      RHOE(1:NDIM1,1:NDIM2) = 0.0
      RHOI(1:NDIM1,1:NDIM2) = 0.0
      AJ1I(1:NDIM1,1:NDIM2) = 0.0
      AJ2I(1:NDIM1,1:NDIM2) = 0.0
      AJ1E(1:NDIM1,1:NDIM2) = 0.0
      AJ2E(1:NDIM1,1:NDIM2) = 0.0
      E1(1:NDIM1,1:NDIM2)=1.E8
      E2(1:NDIM1,1:NDIM2)=1.E8
      B3(1:NDIM1,1:NDIM2)=1.
C
      WRITE(6,2000)
      READ(5,*)DT,NDT,NDTE,NNDTE,NPSORT
      WRITE(6,2040) DT,NDT,NDTE,NNDTE,NPSORT

      READ(5,*)NCOMPR,NPROC,IDEFLT,MESCNT,NPTCLS
      WRITE(6,2070) NCOMPR,NPROC,IDEFLT,MESCNT,NPTCLS

      DTE = DT/(NDTE*NNDTE)

C  DEFINITION OF THE GRID (SQUARE 1 CM X 1 CM WITH 11 x 21 POINTS)
      DO 11 I2=1,I2MX
      DO 11 I1=1,I1MX
        X1(I1,I2)=FLOAT(I1-1)/1000.
        X2(I1,I2)=FLOAT(I2-1)/2000.
   11 CONTINUE

C  COMPUTATION OF INVERSE VOLUMES
      DO 12 I2=2,I2MX-1
        V(1,I2)=4.0E6
        V(I1MX,I2)=4.0E6
        DO 12 I1=2,I1MX-1
          V(I1,I2)=2.0E6
   12 CONTINUE
```

```
      DO 13 I1=2,I1MX-1
        V(I1,1)=4.0E6
        V(I1MX,1)=4.0E6
   13 CONTINUE
      V(1,1)=8.0E6
      V(I1MX,1)=8.0E6
      V(I1MX,I2MX)=8.0E6
      V(1,I2MX)=8.0E6

C  DEFINITION OF THE PARTICLES
C
C  ELECTRONS:
      NPS1=10
      DO 32 I=1,NPS1
      PS1(I,1)=(X1(I1MX,1)-X1(1,1))*I/(NPS1+1)
      PS1(I,2)=(X2(1,I2MX)-X2(1,1))*I/(NPS1+1)
      PS1(I,3)=0.
      PS1(I,4)=0.
      PS1(I,5)=-1.E-10
      PS1(I,6)=(X1(I1MX,1)-X1(1,1))*I/(NPS1+1)
      PS1(I,7)=(X2(1,I2MX)-X2(1,1))*I/(NPS1+1)
   32 CONTINUE

C  ONE NEW ELECTRON IN EACH TIME STEP:
      PS1N(1,1:7)=PS1(10,1:7)
      PS1N(1,5)=-0.15E-09

C  PROTONS:
      NPS2=10
      DO 42 I=1,NPS2
      PS2(I,1)=(X1(I1MX,1)-X1(1,1))*I/(NPS2+1)
      PS2(I,2)=(X2(1,I2MX)-X2(1,1))*I/(NPS2+1)
      PS2(I,3)=0.
      PS2(I,4)=0.
      PS2(I,5)=1.E-10
      PS2(I,6)=(X1(I1MX,1)-X1(1,1))*I/(NPS2+1)
      PS2(I,7)=(X2(1,I2MX)-X2(1,1))*I/(NPS2+1)
   42 CONTINUE

C  LOCALIZATION OF THE PARTICLES:
      CALL PLOCC(PS1,NPS1MX,NPS1,X1,X2,V,NDIM1,NDIM2,
     &      I1MX,I2MX,IELKR1,IELKR2,NDELKR,NELKR,ICPU1,1,1)
      CALL PLOCC(PS2,NPS2MX,NPS2,X1,X2,V,NDIM1,NDIM2,
     &      I1MX,I2MX,IELKR1,IELKR2,NDELKR,NELKR,ICPU1,1,1)
      WRITE(6,1002)
      DO 31 L=1,NPS1
        WRITE(6,1001) 'EL: ',L,(PS1(L,J),J=1,7)
   31 CONTINUE
      WRITE(6,1002)
```

```
      DO 41 L=1,NPS2
        WRITE(6,1001) 'IO: ',L,(PS2(L,J),J=1,7)
   41 CONTINUE

      HAUPT=MYTASKID()
C  CREATE THE PARTICLE PROCESSES
      CALL CRENOD(MXPROC, PROC, HAUPT, NPROC, NSEND, IDEFLT, NPS1MX,
     -      NPS2MX, NPS3MX, NPS4MX, NPS5MX, NPS6MX, NPS1, NPS2, NPS3,
     -      NPS4, NPS5, NPS6, PS1, PS2, PS3, PS4, PS5, PS6, ANZHLE,
     -      ANZHL2, ANZHL3, ANZHL4, ANZHL5, ANZHL6, NDIM1, NDIM2,
     -      I1MX, I2MX, X1, X2, V, EDM1, EDM2, EDM3, EDM4, EDM5, EDM6,
     -      NELKR, DT, DTE, NPSORT,
     -      0, LAENGE, IERROR, MESCNT, 1)
      NSND=INT(ALOG(FLOAT(MAX(1,NPROC)))/ALOG(2.))+1
      IF(IERROR.NE.0) GOTO 400

C ***************************************************************************
C **  HAUPTZEITSCHLEIFE
C **  -----------------
C ***************************************************************************

      DO 350 I = 1,NDT

      WRITE(6,*)
      WRITE(6,*) I,' -TER ZEITSCHRITT'
      WRITE(6,*)'--------------------'
      WRITE(6,*)

      NPSSND=.TRUE.
      IF(I.EQ.NDT) THEN
        ENDE=.TRUE.
      ELSE
        ENDE=.FALSE.
      ENDIF
      IF(MOD(I,2).EQ.0) THEN
        PTSSND=.TRUE.
      ELSE
        PTSSND=.FALSE.
      ENDIF

C   SEND THE FIELDS AND ONE NEW ELECTRON TO THE PARTICLE PROCESSES
      CALL SNEW(MXPROC, PROC, PTSSND, NPSSND, NPROC,
     -      NPS1MN, 1, PS1N, NPS2MX, 0, PS2, NPS3MX, 0,
     -      PS3, NPS4MX, 0, PS4, NPS5MX, 0, PS5, NPS6MX,
     -      0, PS6,  NDTE, NNDTE, ENDE, NCOMPR, NDIM1, NDIM2,
     -      I1MX, I2MX, E1, E2, B3, NSEND, NSND, 1)
      IF(IERROR.NE.0) GOTO 400

C   WAIT FOR NEW DENSITIES FROM THE PARTICLE PROCESSES
```

```
      CALL RDENS(NPROC,NDIM1,NDIM2,I1MX,I2MX,RHOE,AJ1E,AJ2E,RHOI,AJ1I,
     -    AJ2I,RHOEX,AJ1EX,AJ2EX,RHOIX,NSND,AJ1IX,AJ2IX,
     -    Q1,Q2,QAUS,IERROR,1)
      IF(IERROR.NE.0) GOTO 400

C   WAIT FOR PARTICLE DATA FROM THE PARTICLE PROCESSES
      CALL PTCLSR(PROC, MXPROC, PS1, NPS1MX, NPS1A, PS2, NPS2MX,
     -            NPS2A, PS3, NPS3MX, NPS3A, PS4, NPS4MX, NPS4A,
     -            PS5, NPS5MX, NPS5A, PS6, NPS6MX, NPS6A, NPS1,
     -            NPS2, NPS3, NPS4, NPS5, NPS6, NPROC, ANZHLE,
     -            ANZHL2, ANZHL3, ANZHL4, ANZHL5, ANZHL6, LAST,
     -            NDTE, NNDTE, IERROR)
      WRITE(6,*) 'NUMBER OF PARTICLES OUTSIDE: ',NPS1A,NPS2A
      IF(IERROR.NE.0) GOTO 400

      WRITE(6,*)
      FLAST=FLOAT(SUM(LAST(1:NPROC)))
      WRITE(6,*) '====> TEILCHENZAHLEN :'
      WRITE(6,*)
      WRITE(6,1011)
      WRITE(6,1023)
      DO 3400 L=1,NPROC
        WRITE(6,1012) PROC(L),ANZHLE(L),ANZHL2(L),ANZHL3(L),ANZHL4(L),
     -              ANZHL5(L),ANZHL6(L),LAST(L)/FLAST
 3400 CONTINUE
      WRITE(6,1023)
      WRITE(6,1024) NPS1,NPS2,NPS3,NPS4,NPS5,NPS6
      WRITE(6,1025)

C RECEIVE THE PARTICLES
      IF(PTSSND) THEN
         CALL RPTCLS(PROC, MXPROC, PS1, NPS1MX, NPS1X, PS2, NPS2MX,
     -               NPS2X,  PS3, NPS3MX, NPS3X, PS4, NPS4MX, NPS4X,
     -               PS5, NPS5MX, NPS5X, PS6, NPS6MX, NPS6X,
     -               NPS1A, NPS2A, NPS3A, NPS4A, NPS5A, NPS6A, NPROC,
     -               IERROR, PSFULL, 1, 0)
         WRITE(6,1002)
         DO 33 L=1,NPS1
           WRITE(6,1001) 'EL: ',L,(PS1(L,J),J=1,7)
   33    CONTINUE
         WRITE(6,1002)
         DO 43 L=1,NPS2
           WRITE(6,1001) 'IO: ',L,(PS2(L,J),J=1,7)
   43    CONTINUE
         WRITE(6,*)
      ENDIF

C     CHECK THE LOAD-BALANCE IN THE PARTICLE PROCESSES
      IF(I.NE.NDT) THEN
```

```
          CALL MINMAN(MXPROC, PROC, NPROC, IPFUNC, ANZHLE, ANZHL2,
     -         ANZHL3, ANZHL4, ANZHL5, ANZHL6, NPTCLS, IERROR)
          IF(IERROR.NE.0) GOTO 400
        ENDIF

 350  CONTINUE

C*********************************************************************
C**     ENDE DER ZEITSCHLEIFE ******************************************
C*********************************************************************

C OUTPUT OF THE CHARGE DENSITIES

          WRITE(6,*)'CHARGE DENSITY RHOE :'
          WRITE(6,2610) 1,I1MX
          DO 1135 I2=I2MX,1,-1
          WRITE(6,2620) I2,(RHOE(I1,I2),I1=1,I1MX)
 1135     CONTINUE
          WRITE(6,*)
          WRITE(6,*)'CHARGE DENSITY RHOI :'
          WRITE(6,2610) 1,I1MX
          DO 1138 I2=I2MX,1,-1
          WRITE(6,2620) I2,(RHOI(I1,I2),I1=1,I1MX)
 1138     CONTINUE

C ERROR-EXIT
  400 CONTINUE
          WRITE(6,*) 'ERROR CODE : ',IERROR

       STOP

 2610 FORMAT(11X,'I1 = 1 (',I3,' )',I4/)
 2620 FORMAT(' I2 =',I4,2X,1P10E10.2/29(11X,10E10.2/))
 2000 FORMAT(10(' ********',)/' *',78X,'*'/
     &' *',4X,'HAUPTPROGRAMM ZUR DEMONSTRATION DES '
     &        'ARBEITSPAKETES 2D-PLAS  ',14X,'*'/
     &' *',4X,'(FORTBEWEGUNG VON TEILCHEN IN EINEM '
     &        'RECHTECKGITTER)         ',14X,'*'/
     &' *',78X,'*'/' ',10('********',)////)
 2040 FORMAT(' PARAMETERGRUPPE 1: ',
     &'   DT  =',1PE10.2',   NDT    =',I7,',  NDTE   =',I7,','/20X,
     &'   NNDTE  =',I7,',  NPSORT =',I7,'.'//)
 2070 FORMAT(' PARAMETERGRUPPE 2: ',
     &'   NCOMPR  =',I7,',   NPROC  =',I7,
     &',  IDEFLT  =',I7,','/20X
     &'   MESCNT  =',I7,',  NPTCLS  =',I7,'.'//)
 1001 FORMAT(A4,I3,7E10.3)
 1002 FORMAT(/4X,'NR.',' X-COORD',3X,'Y-COORD',1X,'     VX',3X,
     -       '     VY',5X,' CHARGE',1X,' X-WEIGHT',1X,' Y-WEIGHT')
```

```
 1023 FORMAT(80('-'))
 1025 FORMAT(72('-'))
 1011 FORMAT('] PROCESS  ] ELEKTR. ] IONEN 2 ] IONEN 3 ]',
     -        ' IONEN 4 ] IONEN 5 ] IONEN 6 ]  LAST ]')
 1024 FORMAT('] ',10X,I6,3X,']',5(I6,3X,']'))
 1012 FORMAT('] ',I5,3X,' ]',I6,3X,']',5(I6,3X,']'),F5.2,2X,']')

      END
```

---

Output of a run using the previous program:

```
 ********************************************************************************
 *                                                                              *
 *    HAUPTPROGRAMM ZUR DEMONSTRATION DES ARBEITSPAKETES 2D-PLAS                *
 *    (FORTBEWEGUNG VON TEILCHEN IN EINEM RECHTECKGITTER)                       *
 *                                                                              *
 ********************************************************************************



 PARAMETERGRUPPE 1:    DT  =  3.82E-12,   NDT    =      4,  NDTE   =      4,
                       NNDTE  =      5,  NPSORT =      2.


 PARAMETERGRUPPE 2:    NCOMPR  =      0,   NPROC  =      3,  IDEFLT  =      1,
                       MESCNT  =      1,  NPTCLS  =      1.


     NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
EL:    1 0.909E-03 0.909E-03 0.000E+00 0.000E+00-0.100E-09 0.191E+01 0.282E+01
EL:    2 0.182E-02 0.182E-02 0.000E+00 0.000E+00-0.100E-09 0.282E+01 0.464E+01
EL:    3 0.273E-02 0.273E-02 0.000E+00 0.000E+00-0.100E-09 0.373E+01 0.645E+01
EL:    4 0.364E-02 0.364E-02 0.000E+00 0.000E+00-0.100E-09 0.464E+01 0.827E+01
EL:    5 0.455E-02 0.455E-02 0.000E+00 0.000E+00-0.100E-09 0.555E+01 0.101E+02
EL:    6 0.545E-02 0.545E-02 0.000E+00 0.000E+00-0.100E-09 0.645E+01 0.119E+02
EL:    7 0.636E-02 0.636E-02 0.000E+00 0.000E+00-0.100E-09 0.736E+01 0.137E+02
EL:    8 0.727E-02 0.727E-02 0.000E+00 0.000E+00-0.100E-09 0.827E+01 0.155E+02
EL:    9 0.818E-02 0.818E-02 0.000E+00 0.000E+00-0.100E-09 0.918E+01 0.174E+02
EL:   10 0.909E-02 0.909E-02 0.000E+00 0.000E+00-0.100E-09 0.101E+02 0.192E+02

     NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
IO:    1 0.909E-03 0.909E-03 0.000E+00 0.000E+00 0.100E-09 0.191E+01 0.282E+01
IO:    2 0.182E-02 0.182E-02 0.000E+00 0.000E+00 0.100E-09 0.282E+01 0.464E+01
IO:    3 0.273E-02 0.273E-02 0.000E+00 0.000E+00 0.100E-09 0.373E+01 0.645E+01
IO:    4 0.364E-02 0.364E-02 0.000E+00 0.000E+00 0.100E-09 0.464E+01 0.827E+01
IO:    5 0.455E-02 0.455E-02 0.000E+00 0.000E+00 0.100E-09 0.555E+01 0.101E+02
```

```
IO:    6 0.545E-02 0.545E-02 0.000E+00 0.000E+00 0.100E-09 0.645E+01 0.119E+02
IO:    7 0.636E-02 0.636E-02 0.000E+00 0.000E+00 0.100E-09 0.736E+01 0.137E+02
IO:    8 0.727E-02 0.727E-02 0.000E+00 0.000E+00 0.100E-09 0.827E+01 0.155E+02
IO:    9 0.818E-02 0.818E-02 0.000E+00 0.000E+00 0.100E-09 0.918E+01 0.174E+02
IO:   10 0.909E-02 0.909E-02 0.000E+00 0.000E+00 0.100E-09 0.101E+02 0.192E+02

 1  -TER ZEITSCHRITT
 --------------------

 NUMBER OF PARTICLES OUTSIDE: 0 0

 ====> TEILCHENZAHLEN :

| PROCESS  | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 |  LAST |
-------------------------------------------------------------------------------
|     1    |     4   |     3   |     0   |     0   |     0   |     0   | 0.36  |
|     2    |     3   |     3   |     0   |     0   |     0   |     0   | 0.27  |
|     3    |     4   |     4   |     0   |     0   |     0   |     0   | 0.36  |
-------------------------------------------------------------------------------
|              11   |    10   |     0   |     0   |     0   |     0   |
-------------------------------------------------------------------

 2  -TER ZEITSCHRITT
 --------------------

 NUMBER OF PARTICLES OUTSIDE: 0 0

 ====> TEILCHENZAHLEN :

| PROCESS  | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 |  LAST |
-------------------------------------------------------------------------------
|     1    |     5   |     3   |     0   |     0   |     0   |     0   | 0.42  |
|     2    |     3   |     3   |     0   |     0   |     0   |     0   | 0.25  |
|     3    |     4   |     4   |     0   |     0   |     0   |     0   | 0.33  |
-------------------------------------------------------------------------------
|              12   |    10   |     0   |     0   |     0   |     0   |
-------------------------------------------------------------------

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
EL:   1 0.676E-03 0.298E-03-0.253E+08-0.151E+09-0.100E-09 0.168E+01 0.160E+01
EL:   2 0.159E-02 0.121E-02-0.253E+08-0.151E+09-0.100E-09 0.259E+01 0.341E+01
EL:   3 0.249E-02 0.212E-02-0.253E+08-0.151E+09-0.100E-09 0.349E+01 0.523E+01
EL:   4 0.886E-02 0.848E-02-0.253E+08-0.151E+09-0.150E-09 0.986E+01 0.180E+02
EL:   5 0.899E-02 0.894E-02-0.393E+08-0.799E+08-0.150E-09 0.999E+01 0.189E+02
EL:   6 0.340E-02 0.303E-02-0.253E+08-0.151E+09-0.100E-09 0.440E+01 0.705E+01
EL:   7 0.431E-02 0.393E-02-0.253E+08-0.151E+09-0.100E-09 0.531E+01 0.887E+01
EL:   8 0.522E-02 0.484E-02-0.253E+08-0.151E+09-0.100E-09 0.622E+01 0.107E+02
EL:   9 0.613E-02 0.575E-02-0.253E+08-0.151E+09-0.100E-09 0.713E+01 0.125E+02
EL:  10 0.704E-02 0.666E-02-0.253E+08-0.151E+09-0.100E-09 0.804E+01 0.143E+02
```

```
EL:  11 0.795E-02 0.757E-02-0.253E+08-0.151E+09-0.100E-09 0.895E+01 0.161E+02
EL:  12 0.886E-02 0.848E-02-0.253E+08-0.151E+09-0.100E-09 0.986E+01 0.180E+02

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
IO:   1 0.910E-03 0.910E-03 0.732E+05 0.732E+05 0.100E-09 0.191E+01 0.282E+01
IO:   2 0.182E-02 0.182E-02 0.732E+05 0.732E+05 0.100E-09 0.282E+01 0.464E+01
IO:   3 0.273E-02 0.273E-02 0.732E+05 0.732E+05 0.100E-09 0.373E+01 0.646E+01
IO:   4 0.364E-02 0.364E-02 0.732E+05 0.732E+05 0.100E-09 0.464E+01 0.827E+01
IO:   5 0.455E-02 0.455E-02 0.732E+05 0.732E+05 0.100E-09 0.555E+01 0.101E+02
IO:   6 0.545E-02 0.545E-02 0.732E+05 0.732E+05 0.100E-09 0.645E+01 0.119E+02
IO:   7 0.636E-02 0.636E-02 0.732E+05 0.732E+05 0.100E-09 0.736E+01 0.137E+02
IO:   8 0.727E-02 0.727E-02 0.732E+05 0.732E+05 0.100E-09 0.827E+01 0.155E+02
IO:   9 0.818E-02 0.818E-02 0.732E+05 0.732E+05 0.100E-09 0.918E+01 0.174E+02
IO:  10 0.909E-02 0.909E-02 0.732E+05 0.732E+05 0.100E-09 0.101E+02 0.192E+02


 3  -TER ZEITSCHRITT
 --------------------

 NUMBER OF PARTICLES OUTSIDE: 1 0

 ====> TEILCHENZAHLEN :

| PROCESS  | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 |  LAST |
-------------------------------------------------------------------------------
|     1    |     5   |     3   |     0   |     0   |     0   |     0   | 0.38  |
|     2    |     4   |     3   |     0   |     0   |     0   |     0   | 0.31  |
|     3    |     4   |     4   |     0   |     0   |     0   |     0   | 0.31  |
-------------------------------------------------------------------------------
|              13   |    10   |     0   |     0   |     0   |     0   |
-------------------------------------------------------------------

 4  -TER ZEITSCHRITT
 --------------------

 NUMBER OF PARTICLES OUTSIDE: 1 0

 ====> TEILCHENZAHLEN :

| PROCESS  | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 |  LAST |
-------------------------------------------------------------------------------
|     1    |     5   |     3   |     0   |     0   |     0   |     0   | 0.38  |
|     2    |     4   |     3   |     0   |     0   |     0   |     0   | 0.31  |
|     3    |     4   |     4   |     0   |     0   |     0   |     0   | 0.31  |
-------------------------------------------------------------------------------
|              13   |    10   |     0   |     0   |     0   |     0   |
-------------------------------------------------------------------

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
```

```
EL:   1 0.899E-02 0.894E-02-0.393E+08-0.799E+08-0.150E-09 0.999E+01 0.189E+02
EL:   2 0.166E-02-0.232E-04 0.494E+08-0.202E+09-0.100E-09-0.110E+01-0.510E+01
EL:   3 0.262E-02 0.692E-03 0.606E+08-0.203E+09-0.100E-09 0.362E+01 0.238E+01
EL:   4 0.899E-02 0.706E-02 0.606E+08-0.203E+09-0.150E-09 0.999E+01 0.151E+02
EL:   5 0.886E-02 0.848E-02-0.253E+08-0.151E+09-0.150E-09 0.986E+01 0.180E+02
EL:   6 0.353E-02 0.160E-02 0.606E+08-0.203E+09-0.100E-09 0.453E+01 0.420E+01
EL:   7 0.444E-02 0.251E-02 0.606E+08-0.203E+09-0.100E-09 0.544E+01 0.602E+01
EL:   8 0.535E-02 0.342E-02 0.606E+08-0.203E+09-0.100E-09 0.635E+01 0.784E+01
EL:   9 0.884E-02 0.782E-02 0.151E+08-0.190E+09-0.150E-09 0.984E+01 0.166E+02
EL:  10 0.626E-02 0.433E-02 0.606E+08-0.203E+09-0.100E-09 0.726E+01 0.966E+01
EL:  11 0.717E-02 0.524E-02 0.606E+08-0.203E+09-0.100E-09 0.817E+01 0.115E+02
EL:  12 0.808E-02 0.615E-02 0.606E+08-0.203E+09-0.100E-09 0.908E+01 0.133E+02
EL:  13 0.899E-02 0.706E-02 0.606E+08-0.203E+09-0.100E-09 0.999E+01 0.151E+02

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
IO:   1 0.910E-03 0.910E-03 0.146E+06 0.146E+06 0.100E-09 0.191E+01 0.282E+01
IO:   2 0.182E-02 0.182E-02 0.146E+06 0.146E+06 0.100E-09 0.282E+01 0.464E+01
IO:   3 0.273E-02 0.273E-02 0.146E+06 0.146E+06 0.100E-09 0.373E+01 0.646E+01
IO:   4 0.364E-02 0.364E-02 0.146E+06 0.146E+06 0.100E-09 0.464E+01 0.828E+01
IO:   5 0.455E-02 0.455E-02 0.146E+06 0.146E+06 0.100E-09 0.555E+01 0.101E+02
IO:   6 0.546E-02 0.546E-02 0.146E+06 0.146E+06 0.100E-09 0.646E+01 0.119E+02
IO:   7 0.637E-02 0.637E-02 0.146E+06 0.146E+06 0.100E-09 0.737E+01 0.137E+02
IO:   8 0.727E-02 0.727E-02 0.146E+06 0.146E+06 0.100E-09 0.827E+01 0.155E+02
IO:   9 0.818E-02 0.818E-02 0.146E+06 0.146E+06 0.100E-09 0.918E+01 0.174E+02
IO:  10 0.909E-02 0.909E-02 0.146E+06 0.146E+06 0.100E-09 0.101E+02 0.192E+02

 CHARGE DENSITY RHOE+RHOI :
           I1 = 1 (  1 )  11
 I2 =  21    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
 I2 =  20    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  0.00E+00  0.00E+00  3.35E-05  6.81E-06
 I2 =  19    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  0.00E+00 -1.76E-06 -1.11E-04  3.01E-05
 I2 =  18    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  0.00E+00  1.87E-05 -2.72E-04  0.00E+00
 I2 =  17    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  0.00E+00  7.12E-05 -1.46E-04  0.00E+00
 I2 =  16    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  7.96E-05  1.14E-05 -1.48E-04  0.00E+00
 I2 =  15    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             0.00E+00  6.56E-05  1.92E-05 -4.39E-04  0.00E+00
 I2 =  14    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             9.27E-05  5.33E-05 -5.40E-05 -4.60E-06  0.00E+00
 I2 =  13    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
             3.43E-05  1.97E-05 -1.30E-04 -1.11E-05  0.00E+00
 I2 =  12    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  9.92E-05
             8.32E-05 -7.89E-05 -1.61E-05  0.00E+00  0.00E+00
 I2 =  11    0.00E+00  0.00E+00  0.00E+00  0.00E+00  8.49E-06  1.98E-05
```

```
              8.04E-06 -8.73E-05 -1.78E-05  0.00E+00  0.00E+00
I2 =  10      0.00E+00  0.00E+00  0.00E+00  0.00E+00  8.21E-05  9.91E-05
             -9.71E-05 -3.42E-05  0.00E+00  0.00E+00  0.00E+00
I2 =   9      0.00E+00  0.00E+00  0.00E+00  2.00E-05  3.51E-05  0.00E+00
             -5.08E-05 -1.79E-05  0.00E+00  0.00E+00  0.00E+00
I2 =   8      0.00E+00  0.00E+00  0.00E+00  5.25E-05  9.24E-05 -1.09E-04
             -5.89E-05  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   7      0.00E+00  0.00E+00  2.48E-05  6.66E-05 -2.27E-06 -2.28E-05
             -1.13E-05  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   6      0.00E+00  0.00E+00  2.94E-05  7.91E-05 -1.09E-04 -8.66E-05
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   5      0.00E+00  2.31E-05  1.05E-04 -1.89E-05 -2.16E-05  0.00E+00
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   4      0.00E+00  1.30E-05  5.91E-05 -7.45E-05 -8.51E-05  0.00E+00
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   3      2.94E-05  1.49E-04 -2.89E-05 -4.79E-05  0.00E+00  0.00E+00
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   2      6.41E-06  3.26E-05 -4.63E-05 -7.69E-05  0.00E+00  0.00E+00
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   1      0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
              0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
ERROR CODE : 0
```

# 4. Algorithms

The following description of the individual modules is mainly from [13] :

## 4.1 Interpolation

If a particle $P(\alpha_1,\alpha_2)$ is located in cell (i,j), the field $E_p$ at the particle position is calculated from the fields $E_{i,j}$, $E_{i+1,j}$, $E_{i,j+1}$, $E_{i+1,j+1}$ given at the mesh points by use of the area-weighting method [cf. 1,2]:

$$E_p = (1-\alpha_1)(1-\alpha_2)E_{i,j} + \alpha_1(1-\alpha_2)E_{i+1,j} + (1-\alpha_1)\alpha_2 E_{i,j+1} + \alpha_1\alpha_2 E_{i+1,j+1}.$$

In order to be able to apply the standard area-weighting method in an arbitrary quadrangle Q the non-equidistant cell is transformed into a unit square $I^2$.

If $(x,y) \in Q$ is the position of the particle, the weights $(\alpha_1,\alpha_2) \in I^2$ are calculated by the following scheme (see [6,7]):

$$\alpha_2 = \frac{-p+\sqrt{p^2+q}}{(x^s_{i+1,j+1}-1)} \quad for\ x^s_{i+1,j+1} \neq 1,$$

$$\alpha_2 = \frac{y^s}{1+x^s(y^s_{i+1,j+1}-1)} \quad for\ x^s_{i+1,j+1} = 1,$$

$$\alpha_1 = \frac{x^s}{1+\alpha_2(x^s_{i+1,j+1}-1)},$$

$$where \begin{pmatrix} x^s \\ y^s \end{pmatrix} := \begin{pmatrix} x_{i+1,j}-x_{i,j} & x_{i,j+1}-x_{i,j} \\ y_{i+1,j}-y_{i,j} & y_{i,j+1}-y_{i,j} \end{pmatrix}^{-1} \begin{pmatrix} x-x_{i,j} \\ y-y_{i,j} \end{pmatrix},$$

$$p = \tfrac{1}{2}(1+x^s(y^s_{i+1,j+1}-1)-y^s(x^s_{i+1,j+1}-1)) \text{ and } q = y^s(x^s_{i+1,j+1}-1).$$

## 4.2 Particle Pusher

The condition for the movement of the particles is the existence of fields at the mesh points. The interpolation from the fields given at the mesh points onto the particle position produces the forces acting on the particles.

The relativistic equation of motion of electrically charged particles with charge q in an electric field **E** and magnetic field **B** is given by the relativistic Lorentz equation,

$$\mathbf{F} = \frac{d(m\frac{d\mathbf{x}}{dt})}{dt} = q(\mathbf{E} + \frac{d\mathbf{x}}{dt} \times \mathbf{B})$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \frac{d\mathbf{x}}{dt}(0) = \mathbf{v}_0,$$

$$with \quad m = m_0 \gamma \quad and \quad \gamma = \frac{1}{\sqrt{1 - \|\frac{d\mathbf{x}}{dt}\|^2 / c^2}},$$

$m_0$ being the rest mass, **x** the position, and **v** the velocity of the particle.

Depending on the type of particles, the particles are moved with a relativistic (electrons) or non-relativistic (ions) particle pusher by means of the Boris-algorithm [11]. The length of the time step $\Delta t$ is restricted by

$$(q/m) \cdot |B| \cdot \Delta t \leq 0.2$$

in order to assure sufficient numerical accuracy. As the electrons are much faster than the ions, a sub-time scale is introduced. During one ion time step the electrons are advanced several small time steps (see the description of the variables in chapter 2).

After the particles have been advanced, all particles are localized in the grid. Particles outside the computational area are marked and the address of the cell in which a particle is located is assigned to each particle.

## 4.3 Localization

The localization of the particles inside the grid is performed by iteratively using the extended area-weighting method described above. For details see [12].

## 4.4 Computation of the charge and current densities

As a consequence of the use of a leapfrog scheme in order to solve the equation of motion, the location, **x**, and the velocity, **v**, are not known at the same point of time. The velocity is computed half a time step earlier. Hence, before computing the current densities at the mesh points from the particle coordinates, the velocities of each particle are extrapolated half a time step. With these extrapolated velocities the current densities on the grid are calculated, using the same weights as for the interpolation of the fields onto the positions of the particles. The positions of the particles are already known at the right point of time. These positions of the particles are used to determine the charge density on the grid.

## 5. Programming Details

The parallelization strategy is as follows:

Each process which has to treat particles, receives a certain amount of particles from the controlling process (which may be either the initial process or a node process) and the field strengths given at all the points of the grid. The particle processes interpolate the fields to the particle positions and advance the particles by one time step. The charge of the particles are assigned to the grid points and the computed (partial) densities are sent back to the controlling process.

In order to avoid a bottleneck when the particle processes send their data back communication is carried out using a tree-structure (see fig. 5, where the controlling process runs on node 1, particle process PROC(1) runs on node 2, and so on). The controlling process sends data to the particle processes with the identifications PROC(KK), where

$$KK = 2^{i-1},\ i = 1,\ldots,NSND,\ \text{if } KK \leqq NPROC$$

with $NSND = INT(\log_2(NPROC)) + 1$ being the amount of sends the controlling process has to perform, and NPROC stands for the number of particle processes.
If $MOD(KK,2^i) = 0$, for $i = NSND,1,-1$ ,
then particle process PROC(KK) sends data to another particle process PROC(K), with

$$K = KK + 2^{i-1} \text{ (if } KK \leqq NPROC).$$

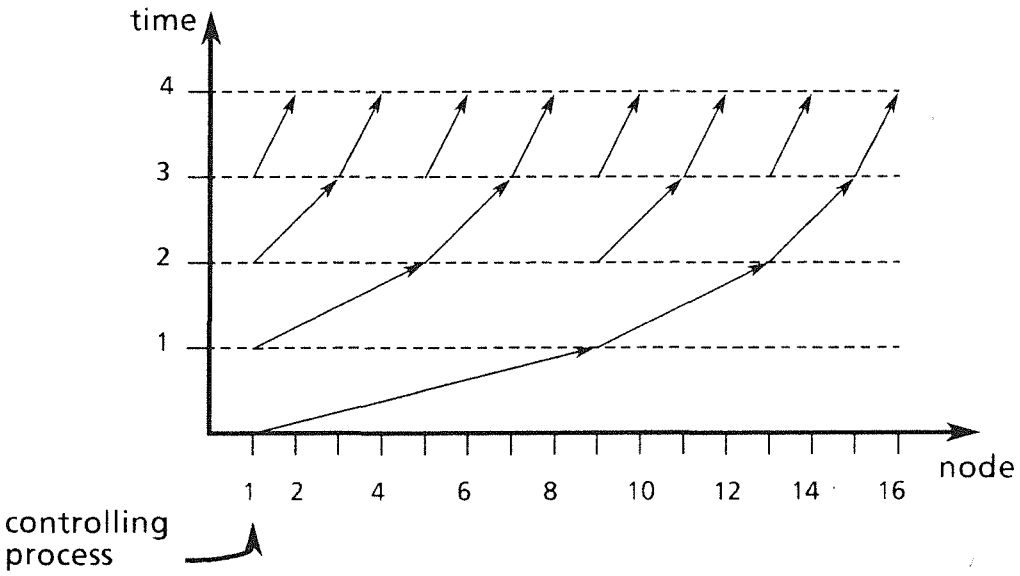


Fig. 5 : sequence of broadcasting (with 15 particle processes): First, process 1 sends data to process 9, which, in the next step sends to process 13, while process 1 sends to process 5, and so on.

## 6. Performance

Not known yet

## 7. References


[1] R.W. Hockney, J.W.Eastwood:
Computer Simulation Using Particles.
McGraw-Hill, 1981

[2] C.K. Birdsall, A.B. Langdon:
Plasma Physics via Computer Simulations.
McGraw-Hill, 1985

[3] T. Westermann:
A Particle-in-Cell Method as a Tool for Diode Simulations.
Nucl. Instr. Meth.A263 (1988), S. 271-279

[4] T. Westermann:
Numerische Simulationen von technisch relevanten Ionen-Dioden mit der Particle-in-Cell Methode.
Kernforschungszentrum Karlsruhe, KfK 4510, Januar 1989

[5] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin:
Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review.
J. Comp. Phys. 47 (1982), 1-108

[6] D. Seldner, T. Westermann:
Numerische Algorithmen für zweidimensionale Teilchen-Simulationsmodelle in technisch relevanten Geometrien.
Kernforschungszentrum Karlsruhe, KfK 4282, June 1987

[7] D. Seldner, T. Westermann:
Algorithms for Interpolation and Localization in Irregular 2D Meshes.
J. Comp. Phys. Vol. 79 No. 1, Nov. 1988, 1-11

[8] T. Westermann:
Teilchen-Fortbewegung in elektro-magnetischen Feldern.
Kernforschungszentrum Karlsruhe, KfK 4325, Januar 1989

[9] M. Alef:
Effiziente Berechnung elektrostatischer Potentiale mit Mehrgittermethoden in technischen Geometrien.
Kernforschungszentrum Karlsruhe, KfK 4613, 1989

[10] D. Seldner:
Modelle zur Parallelisierung der Teilchenbehandlung in Particle-in-Cell Codes auf MIMD-Rechnern mit lokalem Speicher am Beispiel SUPRENUM.
Kernforschungszentrum Karlsruhe, KfK 4495, January 1989

[11] J.P. Boris:
Relativistic Plasma Simulation - Optimization of a Hybrid Code.
Proc. 4. Conf. on Num. Sim. of Plasmas, 3-67, Washington, 1970

[12] Th. Westermann:
Methodology of the BFCPIC Code based on Boundary-Fitted Coordinates and its Application to Ion Diodes.
Proc. Thirteenth Conference on Numerical Simulation of Plasmas, Santa Fé, NM(USA), September 1989, paper IM 3.

[13] Th. Westermann:
Modelling and Simulation of a Diode Relevant for Physical Research.
Proc. 12th IMACS World Congress on Scientific Computation '88, Vol. 3, pp. 244 -246, Paris.

[14] W. K. Giloi:
SUPRENUM: A Trendsetter in Modern Supercomputer Development
Parallel Computing 7 (Sonderheft; U. Trottenberg, Hrsg.), 1988,
pp. 283-296

[15] U. Trottenberg:
On the SUPRENUM Conception (Version 2).
SUPRENUM Report 1, SUPRENUM GmbH, Bonn, Januar 1987

[16] U. Trottenberg:
SUPRENUM - a MIMD system for multilevel scientific supercomputing.
SUPRENUM Report 2, SUPRENUM GmbH, Bonn, Februar 1987

Further References:

E. Halter:
Die Berechnung elektrostatischer Felder in Pulsleistungsanlagen.
Kernforschungszentrum Karlsruhe, KfK 4072, April 1986

M. Alef, D. Seldner, T. Westermann:
Numerische Algorithmen für elektrodynamische Modelle und ihre Implementierung auf Supercomputern.
Informatik-Fachberichte 150 (J. Halin, Hrsg.), Springer-Verlag 1987,
S. 298-305

D. Seldner, M. Alef, T. Westermann, E. Halter:
Parallel Particle Simulation in High Voltage Diodes
(Algorithms and Concepts for Implementation on SUPRENUM).
Parallel Computing 7 (Sonderheft; U. Trottenberg, Hrsg.), 1988,
S. 445-449

M. Alef, D. Grether, D. Seldner, T. Westermann:
Diodensimulation mit der „Particle-in-Cell"-Methode und mögliche Implementierung auf SUPRENUM.
KfK-Nachrichten Jahrgang 20 (1988), S. 179-187

M. Alef:
Concepts for Efficient Implementation of Multigrid Methods on SUPRENUM-Like Architectures.
Kernforschungszentrum Karlsruhe, KfK 4614, 1989

## Acknowledgement

The author would like to thank Prof. Dr. Heinz Trauboth, who made the participation of the Nuclear Research Center in the SUPRENUM project possible. Thanks also to Rainer Vogelsang of the SUPRENUM GmbH for valuable help when implementing the program on the SUPRENUM computer. Especially, I would like thank the author of the scalar BFCPIC code, Thomas Westermann, for technical advice and for the review of the manuscript.

# Appendix

## Program Documentation of the Controlling Process STEUER

**Contents:**

## A.1. Overview

STEUER is a task program developed in order to simplify usage of the BFCPIC-P code. It performs the time step loop and sends diagnostic values to the host program.

As STEUER communicates with the particle processes (which must have been created by the initial process, see subroutine CRENOD in the documentation of 2D-PLAS) all modules belonging to the program package 2D-PLAS must be available as well as the task program FELD (with corresponding subroutines) which is designed to compute the electric fields (see the documentation of 2D-DIO[1]). The corresponding processes are created by STEUER. Besides these two program packages the following modules are called by STEUER:

| | |
|---|---|
| RPDATA | (supplied; receive of initial data) |
| RHOMK1 | (supplied; division of the charge density by $\varepsilon_0$) |
| BFELD | (supplied; computation of magnetic fields) |
| GITADD | (supplied; addition of fields) |
| STROM | (supplied; computation of currents) |
| BRDSTR | (supplied; computation of the magnetic field induced by boundary currents) |
| AIRAND | (supplied; computation of boundary currents) |

In the following the modules needed for usage of STEUER are described (see also the example and the documentations for 2D-PLAS and 2D-DIO)

[1] M. Alef: Parallele Berechnung elektrostatischer Potentiale und Felder in technischen Geometrien auf SUPRENUM -Benutzerhandbuch EPOTZR-P und EFLDZR-P -
to appear as KfK 4688, 1990

## A.2. Usage

SPDNOD creates STEUER and supplies the task with data. STEUER then controls the time step loop of the BFCPIC-P code. Start of a new time step is initiated by ITSTRT, the results are received in ITEND. In this section the usage of the subroutines is described.

### A.2.1. SUBROUTINE SPDNOD

Function:

Creation of the controlling process and send of data to this process

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL SPDNOD(RHOE, RHOI, AJ1E, AJ2E, AJ1I, AJ2I, LISTMX,
     -            NDIM1, NDIM2, IPATT, IFALL, PPAR, X1, X2, V,
     -            VOLT, HAUPT, NPROC, NPSORT, I1MX, I2MX,
     -            NPTCLS, NELKR, NPS1, NPS2, NPS3, NPS4, NPS5, NPS6,
     -            NDT, NDTE, NNDTE, NMINM, NCOMPR, IERROR, DT, DTE,
     -            NPRMX, IFILEI)
```

List of arguments:

| Name | function | type | dimension |
|---|---|---|---|
| RHOE | *Input*<br>electron charge density in the diode | REAL | (NDIM1,NDIM2) |
| RHOI | *Input*<br>ion charge density in the diode | REAL | (NDIM1,NDIM2) |
| AJ1E | *Input*<br>electron current density in 1-direction | REAL | (NDIM1,NDIM2) |
| AJ2E | *Input*<br>electron current density in 2-direction | REAL | (NDIM1,NDIM2) |
| AJ1I | *Input*<br>ion current density in 1-direction | REAL | (NDIM1,NDIM2) |
| AJ2I | *Input*<br>ion current density in 2-direction | REAL | (NDIM1,NDIM2) |
| LISTMX | *Input*<br>maximum number of different attributes for grid points (dimension) | INTEGER | |
| NDIM1 | *Input*<br>maximum number of grid points in 1-direction (dimension) | INTEGER | |
| NDIM2 | *Input*<br>maximum number of grid points in 2-direction (dimension) | INTEGER | |
| IPATT | *Input*<br>Table of "point attributes" assigning a certain value to each grid point (see program documentation of 2D-DIO) | INTEGER | (NDIM1,NDIM2) |
| IFALL | *Input*<br>table of cases of grid points. See documentation of 2D-DIO | INTEGER | (-LISTMX:LISTMX) |

| Name | function | type | dimension |
|---|---|---|---|
| PPAR | *Input* | REAL | (-LISTMX:LISTMX,2) |
| | "point parameters". See documentation of 2D-DIO | | |
| X1 | *Input* | REAL | (NDIM1,NDIM2) |
| | coordinates of grid points in 1-direction | | |
| X2 | *Input* | REAL | (NDIM1,NDIM2) |
| | coordinates of grid points in 2-direction | | |
| V | *Input* | REAL | (NDIM1,NDIM2) |
| | inverse volumes of cells | | |
| VOLT | *Input* | INTEGER | |
| | applied voltage | | |
| HAUPT | *Output* | TASKID | |
| | process identification of the controlling process | | |
| NPROC | *Input* | INTEGER | |
| | actual number of particle processes (see 2D-PLAS) | | |
| NPSORT | *Input* | INTEGER | |
| | number of particle species ($1 \leq$ NPSORT $\leq 6$) | | |
| I1MX | *Input* | INTEGER | |
| | number of grid points in 1-direction | | |
| I2MX | *Input* | INTEGER | |
| | number of grid points in 2-direction | | |
| NPTCLS | *Input* | INTEGER | |
| | corresponds to load-balancing of the particle-processes: if less then NPTCLS particles would be exchanged between two particle-processes, no exchange is performed (see 2D-PLAS) | | |
| NELKR | *Input* | INTEGER | |
| | number of inner elecrode-cells | | |

| Name | function | type | dimension |
|---|---|---|---|
| **NPS*i*** | ***Input*** | **INTEGER** | |
| (*i* = 1,...,6) | number of particles of species *i* in the diode | | |
| **NDT** | ***Input*** | **INTEGER** | |
| | number of time steps to be performed | | |
| **NDTE** | ***Input*** | **INTEGER** | |
| | number of small electron time steps with new localization per large ion time step | | |
| **NNDTE** | ***Input*** | **INTEGER** | |
| | number of small electron time steps before new localization | | |
| **NMINM** | ***Input*** | **INTEGER** | |
| | corresponds to load-balancing of the particle-processes: every NMINM time steps the load-balancing is checked (call of subroutine MINMAN) | | |
| **NCOMPR** | ***Input*** | **INTEGER** | |
| | every NCOMPR time steps the particle matrices are compressed | | |
| **IERROR** | ***Output*** | **INTEGER** | |
| | error code (see A.2.5 or 2.4 for explanation) | | |
| **DT** | ***Input*** | **REAL** | |
| | length of time step | | |
| **DTE** | ***Input*** | **REAL** | |
| | length of electron time step (DTE = DT / (NDTE*NNDTE)) | | |
| **NPRMX** | ***Input*** | **INTEGER** | |
| | maximum number of field processes | | |

| Name | function | type | dimension |
|---|---|---|---|
| IFILEI | *Input* | INTEGER | |

indicates whether this is a completely new run (empty diode)
IFILEI = 0: empty diode
IFILEI >0: there are already particles inside the diode (continuing run)

### A.2.2. SUBROUTINE ITSTRT

Function:

Start of a new time step (send of a corresponding message to the controlling process).

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

none

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

**Usage:**

```
      CALL ITSTRT(HAUPT, ENDE, IFSPHI, PTSSND, IFSNPS, IFSFLD,
     -            IFSRHO, IFSAJ, IFSQ, KSTEP, NDT)
```

List of arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **HAUPT** | ***Input*** | **TASKID** | |
| | identification of the controlling process | | |
| **ENDE** | ***Output*** | **LOGICAL** | |
| | indicates whether this is the last time step<br>ENDE = .TRUE. ⇔ KSTEP = NDT | | |
| **IFSPHI** | ***Input*** | **INTEGER** | |
| | indicates whether the electric potential will be sent back after this time step<br>IFSPHI ≧ 1 : the potential will be sent back (for further information see the description of ITEND)<br>in this version IFSPHI must be set = 0 | | |
| **PTSSND** | ***Input*** | **LOGICAL** | |
| | indicates whether the particle matrices will be sent back after this time step<br>PTSSND = .TRUE. : the particle matrices will be sent back after this time step (for further information see the description of ITEND) | | |
| **IFSNPS** | ***Input*** | **INTEGER** | |
| | indicates whether the number of particles will be sent back after this time step<br>IFSNPS ≧ 1 : the number of particles will be sent back after this time step (for further information see the description of ITEND) | | |
| **IFSFLD** | ***Input*** | **INTEGER** | |
| | indicates whether the fields will be sent back after this time step<br>IFSFLD ≧ 1 : the fields will be sent back (for further information see the description of ITEND) | | |
| **IFSRHO** | ***Input*** | **INTEGER** | |
| | indicates whether the charge densities will be sent back after this time step | | |

| Name | function | type | dimension |
|---|---|---|---|
| | IFSRHO ≧ 1 : the charge densities will be sent back (for further information see the description of ITEND) | | |
| IFSAJ | ***Input*** indicates whether the current densities will be sent back after this time step<br>IFSAJ ≧ 1 : the current densities will be sent back (for further information see the description of ITEND) | **INTEGER** | |
| IFSQ | ***Input*** indicates whether the charge in the diode and other diagnostic values will be sent back after this time step<br>IFSQ ≧ 1 : the data will be sent back (for further information see the description of ITEND) | **INTEGER** | |
| KSTEP | ***Input*** number of this time step | **INTEGER** | |
| NDT | ***Input*** total number of time steps | **INTEGER** | |

### A.2.3. SUBROUTINE ITEND

Function:

Receive of data of a time step from the controlling process.

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

STOPIT (if there is message with TAG = 0 in the mailbox)

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

Usage:

```
      CALL ITEND (KSTEP, MXPROC, PROC, NPROC, HAUPT, LAST, NPS1,
     -            NPS2, NPS3, NPS4, NPS5, NPS6, ANZHLE, ANZHL2,
     -            ANZHL3, ANZHL4, ANZHL5, ANZHL6, NDIM1, NDIM2,
     -            I1MX, I2MX, IFSPHI, IFSFLD, IFSNPS, IFSRHO, IFSAJ,
     -            IFSQ, F, E1, E2, B3, RHOE, AJ1E, AJ2E, RHOI, AJ1I,
     -            AJ2I, Q1, Q2, QAUS, QNEUE, QNEUEL, QNEUFI, QNEUI,
     -            STROME, STROMI, IERRMX, MES)
```

List of arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **KSTEP** | ***Input*** | **INTEGER** | |
| | number of the time step from which data is to be received | | |
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes | | |
| **PROC** | ***Input*** | **TASKID** | **(MXPROC)** |
| | process identifications of the particle processes | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | actual number of particle processes | | |
| **HAUPT** | ***Input*** | **TASKID** | |
| | process identification of the controlling process | | |
| **LAST** | ***Output*** | **INTEGER** | **(MXPROC)** |
| | load balance of the particle processes | | |
| **NPS*i*** | ***Output*** | **INTEGER** | |
| (*i* = 1,...,6) | number of particles of species *i* in the diode (only known if IFSNPS≧ 1) | | |
| **ANZHLE** | ***Output*** | **INTEGER** | **(MXPROC)** |
| | ANZHLE(I) denotes the number of electrons in particle process PROC(I) (only known if IFSNPS ≧ 1) | | |
| **ANZHL*i*** | ***Output*** | **INTEGER** | **(MXPROC)** |
| (*i* = 1,...,6) | ANZHL*i*(I) denotes the number of ions (species *i*) in particle process PROC(I) (only known if IFSNPS ≧ 1) | | |
| **NDIM1** | ***Input*** | **INTEGER** | |
| | maximum number of grid points in 1-direction | | |
| **NDIM2** | ***Input*** | **INTEGER** | |
| | maximum number of grid points in 2-direction | | |

| Name | function | type | dimension |
|---|---|---|---|
| **I1MX** | ***Input*** | **INTEGER** | |
| | number of grid points in 1-direction | | |
| **I2MX** | ***Input*** | **INTEGER** | |
| | number of grid points in 2-direction | | |
| **IFSPHI** | ***Input*** | **INTEGER** | |
| | IFSPHI $\geqq$ 1 : the potential is received | | |
| | IFSPHI $\geqq$ 2 : the potential is printed | | |
| | in this version IFSPHI must be set = 0 | | |
| **IFSFLD** | ***Input*** | **INTEGER** | |
| | IFSFLD $\geqq$ 1 : the fields are received | | |
| | IFSFLD $\geqq$ 2 : the fields are printed | | |
| **IFSNPS** | ***Input*** | **INTEGER** | |
| | IFSNPS $\geqq$ 1 : the number of particles are received | | |
| | IFSNPS $\geqq$ 2 : the number of particles are printed | | |
| | IFSNPS $\geqq$ 4 : the number of particles per process are printed | | |
| **IFSRHO** | ***Input*** | **INTEGER** | |
| | IFSRHO $\geqq$ 1 : the charge densities are received | | |
| | IFSRHO $\geqq$ 2 : the charge densities are printed | | |
| **IFSAJ** | ***Input*** | **INTEGER** | |
| | IFSAJ $\geqq$ 1 : the current densities are received | | |
| | IFSAJ $\geqq$ 2 : the current densities are printed | | |
| **IFSQ** | ***Input*** | **INTEGER** | |
| | IFSQ $\geqq$ 1 : the following quantities are received: Q1, Q2, QAUS, QNEUE, QNEUEL, QNEUFI, QNEUI, STROME, STROMI | | |
| | IFSQ $\geqq$ 2 : the charge in the diode is printed | | |
| **F** | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | potential in the diode (only known if IFSPHI $\geqq$ 1) | | |
| | (not implemented in this version) | | |

| Name | function | type | dimension |
|---|---|---|---|
| E1 | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | electric field in 1-direction (only known if IFSFLD ≧ 1) | | |
| E2 | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | electric field in 2-direction (only known if IFSFLD ≧ 1) | | |
| B3 | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | magnetic field (only known if IFSFLD ≧ 1) | | |
| RHOE | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | electron charge density in the diode (only known if IFSRHO ≧ 1) | | |
| AJ1E | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | electron current density in 1-direction in the diode (only known if IFSAJ ≧ 1) | | |
| AJ2E | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | electron current density in 2-direction in the diode (only known if IFSAJ ≧ 1) | | |
| RHOI | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | ion charge density in the diode (only known if IFSRHO ≧ 1) | | |
| AJ1I | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | ion current density in 1-direction in the diode (only known if IFSAJ ≧ 1) | | |
| AJ2I | ***Output*** | **REAL** | **(NDIM1,NDIM2)** |
| | ion current density in 2-direction in the diode (only known if IFSAJ ≧ 1) | | |
| Q1 | ***Output*** | **REAL** | |
| | electron charge in the diode (only known if IFSQ ≧ 1) | | |
| Q2 | ***Output*** | **REAL** | |
| | ion charge in the diode (only known if IFSQ ≧ 1) | | |

| Name | function | type | dimension |
|---|---|---|---|
| QAUS | *Output* | REAL | |
| | outgoing electron charge (only known if IFSQ ≧ 1) | | |
| QNEUE | *Output* | REAL | |
| | not used possibility for a diagnostic value (only known if IFSQ≧ 1) | | |
| QNEUEL | *Output* | REAL | |
| | not used possibility for a diagnostic value (only known if IFSQ≧ 1) | | |
| QNEUFI | *Output* | REAL | |
| | not used possibility for a diagnostic value (only known if IFSQ≧ 1) | | |
| QNEUI | *Output* | REAL | |
| | not used possibility for a diagnostic value (only known if IFSQ≧ 1) | | |
| STROME | *Output* | REAL | |
| | electron current (only known if IFSQ ≧ 1) | | |
| STROMI | *Output* | REAL | |
| | ion current (only known if IFSQ ≧ 1) | | |
| IERRMX | *Output* | INTEGER | |
| | error code (see A.2.5 or 2.4 for explanation) | | |
| MES | *Input* | INTEGER | |
| | parameter for output in STOPIT | | |

### A.2.4. SUBROUTINE ITACK

Function:

Receive the acknowledgement from the controlling process that all processes have terminated and of remaining messages from the node processes (optional).

Programming language:

SUPRENUM-FORTRAN

Subroutine calls:

STOPIT (if there is message with TAG = 0 in the mailbox)

Author:

David Seldner
Kernforschungszentrum Karlsruhe, IDT
Postfach 3640
D-7500 Karlsruhe 1

**Usage:**

```
CALL ITACK (PROC, MXPROC, NPROC, HAUPT, MES, IERROR)
```

List of arguments:

| Name | function | type | dimension |
|---|---|---|---|
| **PROC** | ***Input*** | **TASKID** | **(MXPROC)** |
| | identification of the particle processes | | |
| **MXPROC** | ***Input*** | **INTEGER** | |
| | maximum number of particle processes (dimension) | | |
| **NPROC** | ***Input*** | **INTEGER** | |
| | number of particle processes | | |
| **HAUPT** | ***Input*** | **TASKID** | |
| | identification of the controlling process | | |
| **MES** | ***Input*** | **INTEGER** | |
| | parameter for the output in STOPIT | | |
| **IERROR** | ***Output*** | **INTEGER** | |
| | highest error code (see A.2.5 or 2.4 for explanation) | | |

### A.2.5 List of error codes

| Number | task program (and subroutine), reason | user's action |
|---|---|---|
| 0 | no error | none |
| 1001 | **STEUER (RPDATA)**<br>NPROC > MXPROC | increase MXPROC (or decrease NPROC) |
| 1111 | **STEUER (main program)**<br>synchronization between master and control process | synchronize master and control process |
| 1112 | **STEUER (main program)**<br>not as many field processes as desired could be created (no termination) | none |
| 1113 | **STEUER (SNEW)**<br>NPROC = 0 | |
| 1213 | **STEUER (RPDATA)**<br>I1MX > NDIM1 or I2MX > NDIM2 | increase NDIM1 and/or NDIM2 |
| 3001 | **STEUER (SNEW)**<br>not all new particles were sent (internal error) | none |

## A.3. Computational Example

The described main program was developed in order perform the same computations as in the example for 2D-PLAS. First, the controlling process os created (by calling SPDNOD) and afterwards the particle processes (CRENOD). Note that when a time step I (I = 2,3,4) has been initiated, the results from the previous time step I-1 are received.

```
      PROGRAM BFCHST
C
C*********************************************************************
C******    HAUPTPROGRAMM DES BFCPIC-CODES  ***************************
C*********************************************************************
C *
C *   BEISPIEL-HAUPTPROGRAMM ZUR DEMONSTRATION DES TASK PROGRAMS
C *   STEUER, DAS DIE ZEITSCHLEIFE EINES PARTICLE-IN-CELL CODES
C *   DURCHFUEHRT
C *
C *   +++++++++++++++ PARALLELVERSION ++++++++++++++++++++++
C *
C *       AUTOR: D. SELDNER
C *                 KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH
C *                 INSTITUT FUER DATENVERARBEITUNG IN DER TECHNIK
C *                 TEL. 5595
C *                 STAND: 11.12.1989
C *       BASIEREND AUF DER GRUNDLAGE EINES PROGRAMMS VON
C *                 TH. WESTERMANN
C *                 KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH
C *                 ABTEILUNG FUER NUMERISCHE PHYSIK, HDI-3
C *                 TEL. 3008
C *
C*********************************************************************
C
      PARAMETER (NDIM1=41,NDIM2=65)
      PARAMETER (NPS1MX=16000,NPS2MX=16000,NPS3MX=1,NPS4MX=1)
      PARAMETER (NPS5MX=1,NPS6MX=1)
      PARAMETER (LISTMX=10)
C
      REAL X1(NDIM1,NDIM2),X2(NDIM1,NDIM2),V(NDIM1,NDIM2)
      INTEGER   IPATT(NDIM1,NDIM2),IFALL(-LISTMX:LISTMX)
      REAL PPAR(-LISTMX:LISTMX,2)
C
      REAL F(NDIM1,NDIM2)
      REAL RHOE(NDIM1,NDIM2),RHOI(NDIM1,NDIM2)
      REAL AJ1E(NDIM1,NDIM2),AJ2E(NDIM1,NDIM2)
      REAL AJ1I(NDIM1,NDIM2),AJ2I(NDIM1,NDIM2)
      REAL E1(NDIM1,NDIM2),E2(NDIM1,NDIM2),B3(NDIM1,NDIM2)
C
```

```
      REAL PS1(NPS1MX,7),PS2(NPS2MX,7),PS3(NPS3MX,7)
      REAL PS4(NPS1MX,7),PS5(NPS2MX,7),PS6(NPS3MX,7)
C
      PARAMETER (MXPROC=64,LAENGE=1000)
      INTEGER ANZHLE(MXPROC),ANZHL2(MXPROC),ANZHL3(MXPROC)
      INTEGER ANZHL4(MXPROC),ANZHL5(MXPROC),ANZHL6(MXPROC)
      INTEGER LAST(MXPROC)
      TASKID HAUPT,PROC(MXPROC)
      LOGICAL ENDE,PSFULL

      DATA  IFILEG,IFILEI,IFILEO /3*0/
      DATA  DT,NDT,NDTE,NNDTE              /0.0,3*0/

      DATA NCOMPR,NPROC,IDEFLT,MESCNT,NPRMX,ITRCE/10,1,3,0,0,0/
      DATA ITRCET,NPTCLS,NMINM/0,10,0/
      DATA ANZHLE,ANZHL2,ANZHL3/MXPROC*0,MXPROC*0,MXPROC*0/
      DATA ANZHL4,ANZHL5,ANZHL6/MXPROC*0,MXPROC*0,MXPROC*0/
      DATA ENDE,IERROR/.FALSE.,0/
      DATA ISIM/1/

      DATA EDM1,EDM2,EDM3,EDM4/-1.7588E11,9.579E7,9.579E7,9.579E7/
      DATA EDM5,EDM6/9.579E7,9.579E7/
C
C     EINLESEN DER PARAMETER ZUR STEUERUNG UND RECHNUNG
C
      WRITE(6,2000)
      READ(5,*) IFILEG,IFILEI,IFILEO
      WRITE(6,2010) IFILEG,IFILEI,IFILEO
      READ(5,*)DT,NDT,NDTE,NNDTE,NPSORT
      WRITE(6,2040) DT,NDT,NDTE,NNDTE,NPSORT

      READ(5,*)NCOMPR,NPROC,IDEFLT,MESCNT,NPRMX,ITRCE
      WRITE(6,2070) NCOMPR,NPROC,IDEFLT,MESCNT,NPRMX,ITRCE
      READ(5,*)ITRCET,NPTCLS,NMINM,POT
      CALL TRACE(ITRCE)

      DTE=DT/(NDTE*NNDTE)
C  DEFINITION OF THE GRID (SQUARE) AND OF THE
C  PARTICLES AS IN EXAMPLE 2D-PLAS
      CALL INPUTG(X1,X2,V,NDIM1,NDIM2,...)
      CALL INPUTP(PS1,PS2,NPS1MX,NPS2MX,...)

      WRITE(6,1002)
      DO 31 L=1,NPS1
        WRITE(6,1001) 'EL: ',L,(PS1(L,J),J=1,7)
   31 CONTINUE
      WRITE(6,1002)
      DO 41 L=1,NPS2
        WRITE(6,1001) 'IO: ',L,(PS2(L,J),J=1,7)
```

```
   41 CONTINUE

C CREATION OF THE TASKS
      CALL SPDNOD(RHOE,RHOI,AJ1E,AJ2E, AJ1I, AJ2I, LISTMX,
     -       NDIM1, NDIM2, IPATT, IFALL, PPAR, X1, X2, V,
     -       VOLT, HAUPT, NPROC, NPSORT, I1MX, I2MX, ITRCE,
     -       NPTCLS, NELKR, NPS1, NPS2, NPS3, NPS4, NPS5, NPS6,
     -       NDT, NDTE, NNDTE, NMINM, NCOMPR,
     -       IERROR, DT, DTE, NPRMX, IFILEI, ISIM)

      CALL CRENOD(MXPROC, PROC, HAUPT, NPROC, LAST, IDEFLT, NPS1MX,
     -      NPS2MX, NPS3MX, NPS4MX, NPS5MX, NPS6MX, NPS1, NPS2, NPS3,
     -      NPS4, NPS5, NPS6, PS1, PS2, PS3, PS4, PS5, PS6, ANZHLE,
     -      ANZHL2, ANZHL3, ANZHL4, ANZHL5, ANZHL6, NDIM1, NDIM2,
     -      I1MX, I2MX, X1, X2, V, EDM1, EDM2, EDM3, EDM4, EDM5, EDM6,
     -      NELKR, DT, DTE, NPSORT,ITRCET, LAENGE, IERROR, MESCNT, ISIM)
      IF(IERROR.GT.0) GOTO 400

C ***************************************************************************
C **  HAUPTZEITSCHLEIFE DES BFCPIC-CODES
C **  ------------------------------------------------
C ***************************************************************************

      IF (NDT.EQ.0) GOTO 400

      DO 350 I = 1,NDT

C**  START TIME STEP I

C  IF THIS IS THE LAST STEP SEND PARTICLES AND DENSITIES BACK
      IF(I.NE.NDT) THEN
        CALL ITSTRT(HAUPT, ENDE, 0, .FALSE., 4, 0, 0, 0, 1, I, NDT)
      ELSE
        CALL ITSTRT(HAUPT, ENDE, 0, .TRUE., 4, 0, 3, 0, 1, I, NDT)
      ENDIF

      IF(I.EQ.1) GOTO 360

      WRITE(6,*)
      WRITE(6,*) I-1,' -TER ZEITSCHRITT'
      WRITE(6,*)'----------------------'
      WRITE(6,*)

C**  RECEIVE DATA FROM TIME STEP I-1

      CALL ITEND(I-1, MXPROC, PROC, NPROC, HAUPT,LAST,NPS1,NPS2,NPS3,
     -      NPS4, NPS5, NPS6, ANZHLE, ANZHL2, ANZHL3, ANZHL4, ANZHL5,
     -      ANZHL6, NDIM1, NDIM2, I1MX, I2MX, 0, 0, 4,
     -      0, 0, 1, F, E1, E2, B3, RHOE,AJ1E,AJ2E,RHOI,
```

```
     -      AJ1I,AJ2I, Q1, Q2, QAUS, QNEUE, QNEUEL,
     -      QNEUFI, QNEUI, STROME, STROMI, IERROR, MESCNT)
       IF(IERROR.GT.0) THEN
          WRITE(6,*) 'HOECHSTER FEHLERCODE : ',IERROR
          WRITE(6,*) 'STOP DES PROGRAMMS'
          STOP
       ENDIF

  360 CONTINUE

  350  CONTINUE

C*************************************************************************
C**     ENDE DER ZEITSCHLEIFE ************************************
C*************************************************************************

       WRITE(6,*)
       WRITE(6,*) NDT,' -TER ZEITSCHRITT'
       WRITE(6,*)'----------------------'
       WRITE(6,*)
       WRITE(6,*)

C**  RECEIVE DATA FROM TIME STEP NDT
           CALL ITEND(NDT,MXPROC,PROC, NPROC, HAUPT,LAST, NPS1, NPS2,NPS3,
     -       NPS4, NPS5, NPS6, ANZHLE, ANZHL2, ANZHL3, ANZHL4, ANZHL5,
     -       ANZHL6, NDIM1, NDIM2, I1MX, I2MX, 0, 0, 4,
     -       3, 0, 1, F, E1, E2, B3, RHOE,AJ1E,AJ2E,RHOI,
     -       AJ1I, AJ2I, Q1, Q2, QAUS, QNEUE, QNEUEL,
     -       QNEUFI, QNEUI, STROME, STROMI, IERROR, MESCNT)
          IF(IERROR.GT.0) THEN
            WRITE(6,*) 'HOECHSTER FEHLERCODE : ',IERROR
            GOTO 400
          ENDIF
           CALL RPTCLS(PROC, MXPROC, PS1, NPS1MX, NPS1X, PS2, NPS2MX,
     -                 NPS2X,  PS3, NPS3MX, NPS3X, PS4, NPS4MX, NPS4X,
     -                 PS5, NPS5MX, NPS5X, PS6, NPS6MX, NPS6X,
     -                 NPS1A, NPS2A, NPS3A, NPS4A, NPS5A, NPS6A, NPROC,
     -                 IERROR, PSFULL, 1, IFILEO, ISIM)
       WRITE(6,1002)
       DO 51 L=1,NPS1
         WRITE(6,1001) 'EL: ',L,(PS1(L,J),J=1,7)
    51 CONTINUE
       WRITE(6,1002)
       DO 61 L=1,NPS2
         WRITE(6,1001) 'IO: ',L,(PS2(L,J),J=1,7)
    61 CONTINUE

C** ACKNOWLEDGEMENT (ALL PARTICLE PROCESSES HAVE TERMINATED)
```

```
      CALL ITACK(PROC,MXPROC,NPROC,HAUPT,MESCNT,IERROR)
      IF(IERROR.GT.0) THEN
         WRITE(6,*) 'HOECHSTER FEHLERCODE : ',IERROR
         WRITE(6,*) 'STOP DES PROGRAMMS'
         STOP
      ENDIF
  400 CONTINUE

      WRITE(6,*) 'ERROR CODE : ',IERROR
      STOP

 1001 FORMAT(A4,I3,7E10.3)
 1002 FORMAT(/4X,'NR.',' X-COORD',3X,'Y-COORD',1X,'     VX',3X,
     -       '     VY',5X,' CHARGE',1X,' X-WEIGHT',1X,' Y-WEIGHT')

 2000 FORMAT('1',10('********',)/' *',78X,'*'/' *',78X,'*'/
     &' *',4X,'SIMULATION EINER HOCHSTROMDIODE MIT EINE'
     &       'M PARTICLE-IN-CELL-VERFAHREN  ',4X,'*'/
     &' *',4X,'AUF EINEM GITTER MIT RANDANGEPASSTEN KOORDINATEN. '
     &       '                    ',4X,'*'/
     &' *',4X,'                                      '
     &       '                                       ',4X,'*'/
     &' *',78X,'*'/' *',78X,'*'/ ',10('********',)////)
 2010 FORMAT(' PARAMETERGRUPPE 1: ',
     &'   IFILEG =',I7,',   IFILEI =',I7,',   IFILEO =',I7,'.   '//)
 2040 FORMAT(' PARAMETERGRUPPE 2: ',
     &'   DT  =',1PE10.2',   NDT    =',I7,',  NDTE   =',I7,','/20X,
     &'   NNDTE  =',I7,',   NPSORT  =',I7,'.'//)
 2070 FORMAT(' PARAMETERGRUPPE 3: ',
     &'   NCOMPR  =',I7,',   NPROC  =',I7,
     &',  IDEFLT  =',I7,','/20X
     &'   MESCNT  =',I7,',  NPRMX   =',I7,',   ITRCE  =',I7,'.'//)

      END
```

Output:

```
*******************************************************************************
*                                                                             *
*                                                                             *
*    SIMULATION EINER HOCHSTROMDIODE MIT EINEM PARTICLE-IN-CELL-VERFAHREN      *
*    AUF EINEM GITTER MIT RANDANGEPASSTEN KOORDINATEN.                        *
*                                                                             *
*                                                                             *
*                                                                             *
```

```
**************************************************************************************
```

```
 PARAMETERGRUPPE 1:    IFILEG =     20,   IFILEI =      0,   IFILEO =      0.

 PARAMETERGRUPPE 2:    DT  =  3.82E-12,   NDT    =      4,  NDTE   =      4,
                       NNDTE  =      5,   NPSORT  =      2.

 PARAMETERGRUPPE 3:    NCOMPR  =      1,   NPROC  =      3,  IDEFLT  =      1,
                       MESCNT  =      3,   NPRMX  =      1,   ITRCE  =      1.
```

| | NR. | X-COORD | Y-COORD | VX | VY | CHARGE | X-WEIGHT | Y-WEIGHT |
|---|---|---|---|---|---|---|---|---|
| EL: | 1 | 0.909E-03 | 0.909E-03 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.191E+01 | 0.282E+01 |
| EL: | 2 | 0.182E-02 | 0.182E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.282E+01 | 0.464E+01 |
| EL: | 3 | 0.273E-02 | 0.273E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.373E+01 | 0.645E+01 |
| EL: | 4 | 0.364E-02 | 0.364E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.464E+01 | 0.827E+01 |
| EL: | 5 | 0.455E-02 | 0.455E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.555E+01 | 0.101E+02 |
| EL: | 6 | 0.545E-02 | 0.545E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.645E+01 | 0.119E+02 |
| EL: | 7 | 0.636E-02 | 0.636E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.736E+01 | 0.137E+02 |
| EL: | 8 | 0.727E-02 | 0.727E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.827E+01 | 0.155E+02 |
| EL: | 9 | 0.818E-02 | 0.818E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.918E+01 | 0.174E+02 |
| EL: | 10 | 0.909E-02 | 0.909E-02 | 0.000E+00 | 0.000E+00 | -0.100E-09 | 0.101E+02 | 0.192E+02 |

| | NR. | X-COORD | Y-COORD | VX | VY | CHARGE | X-WEIGHT | Y-WEIGHT |
|---|---|---|---|---|---|---|---|---|
| IO: | 1 | 0.909E-03 | 0.909E-03 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.191E+01 | 0.282E+01 |
| IO: | 2 | 0.182E-02 | 0.182E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.282E+01 | 0.464E+01 |
| IO: | 3 | 0.273E-02 | 0.273E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.373E+01 | 0.645E+01 |
| IO: | 4 | 0.364E-02 | 0.364E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.464E+01 | 0.827E+01 |
| IO: | 5 | 0.455E-02 | 0.455E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.555E+01 | 0.101E+02 |
| IO: | 6 | 0.545E-02 | 0.545E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.645E+01 | 0.119E+02 |
| IO: | 7 | 0.636E-02 | 0.636E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.736E+01 | 0.137E+02 |
| IO: | 8 | 0.727E-02 | 0.727E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.827E+01 | 0.155E+02 |
| IO: | 9 | 0.818E-02 | 0.818E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.918E+01 | 0.174E+02 |
| IO: | 10 | 0.909E-02 | 0.909E-02 | 0.000E+00 | 0.000E+00 | 0.100E-09 | 0.101E+02 | 0.192E+02 |

```
 1  -TER ZEITSCHRITT
 ----------------------

 ====> TEILCHENZAHLEN :
```

| PROCESS | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 | LAST |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 4 | 3 | 0 | 0 | 0 | 0 | 0.36 |
| 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0.27 |
| 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0.36 |
| | 11 | 10 | 0 | 0 | 0 | 0 | |

2 -TER ZEITSCHRITT

====> TEILCHENZAHLEN :

| PROCESS | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 | LAST |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 3 | 0 | 0 | 0 | 0 | 0.42 |
| 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0.25 |
| 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0.33 |
| | 12 | 10 | 0 | 0 | 0 | 0 | |

3 -TER ZEITSCHRITT

====> TEILCHENZAHLEN :

| PROCESS | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 | LAST |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 3 | 0 | 0 | 0 | 0 | 0.38 |
| 3 | 4 | 3 | 0 | 0 | 0 | 0 | 0.31 |
| 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0.31 |
| | 13 | 10 | 0 | 0 | 0 | 0 | |

4 -TER ZEITSCHRITT

AUSGABE DER DICHTEN

LADUNGSDICHTEN RHOE :

```
          I1 = 1 (  1 )  11

I2 =  21    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =  20    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =  19    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -1.76E-06 -2.60E-04  0.00E+00
I2 =  18    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -4.11E-05 -2.85E-04  0.00E+00
I2 =  17    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -3.23E-05 -1.69E-04  0.00E+00
I2 =  16    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -1.86E-05 -1.48E-04  0.00E+00
I2 =  15    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -5.53E-06 -4.39E-04  0.00E+00
I2 =  14    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -5.40E-05 -4.60E-06  0.00E+00
I2 =  13    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00 -1.30E-04 -1.11E-05  0.00E+00
I2 =  12    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00 -7.89E-05 -1.61E-05  0.00E+00  0.00E+00
I2 =  11    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00 -8.73E-05 -1.78E-05  0.00E+00  0.00E+00
I2 =  10    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
           -9.71E-05 -3.42E-05  0.00E+00  0.00E+00  0.00E+00
I2 =   9    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
           -5.08E-05 -1.79E-05  0.00E+00  0.00E+00  0.00E+00
I2 =   8    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00 -1.09E-04
           -5.89E-05  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   7    0.00E+00  0.00E+00  0.00E+00  0.00E+00 -2.27E-06 -2.28E-05
           -1.13E-05  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   6    0.00E+00  0.00E+00  0.00E+00  0.00E+00 -1.09E-04 -8.66E-05
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   5    0.00E+00  0.00E+00  0.00E+00 -1.89E-05 -2.16E-05  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   4    0.00E+00  0.00E+00  0.00E+00 -7.45E-05 -8.51E-05  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   3    0.00E+00  0.00E+00 -2.89E-05 -4.79E-05  0.00E+00  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   2    0.00E+00  0.00E+00 -4.63E-05 -7.69E-05  0.00E+00  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
I2 =   1    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
            0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00

LADUNGSDICHTEN RHOI :
          I1 = 1 (  1 )  11

I2 =  21    0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00
```

| | | | | | | |
|---|---|---|---|---|---|---|
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 20 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 3.35E-05 | 6.81E-06 | |
| I2 = 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 1.48E-04 | 3.01E-05 | |
| I2 = 18 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 5.99E-05 | 1.34E-05 | 0.00E+00 | |
| I2 = 17 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 1.04E-04 | 2.32E-05 | 0.00E+00 | |
| I2 = 16 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 7.96E-05 | 3.01E-05 | 0.00E+00 | 0.00E+00 | |
| I2 = 15 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 6.56E-05 | 2.48E-05 | 0.00E+00 | 0.00E+00 | |
| I2 = 14 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 9.27E-05 | 5.33E-05 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 13 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 3.43E-05 | 1.97E-05 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 12 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 9.92E-05 |
| | 8.32E-05 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 11 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 8.49E-06 | 1.98E-05 |
| | 8.04E-06 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 10 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 8.21E-05 | 9.91E-05 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 9 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 2.00E-05 | 3.51E-05 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 8 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 5.25E-05 | 9.24E-05 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 7 | 0.00E+00 | 0.00E+00 | 2.48E-05 | 6.66E-05 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 6 | 0.00E+00 | 0.00E+00 | 2.94E-05 | 7.91E-05 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 5 | 0.00E+00 | 2.31E-05 | 1.05E-04 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 4 | 0.00E+00 | 1.30E-05 | 5.91E-05 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 3 | 2.94E-05 | 1.49E-04 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 2 | 6.41E-06 | 3.26E-05 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |
| I2 = 1 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | |

====> TEILCHENZAHLEN :

| PROCESS | ELEKTR. | IONEN 2 | IONEN 3 | IONEN 4 | IONEN 5 | IONEN 6 | LAST |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 3 | 0 | 0 | 0 | 0 | 0.38 |
| 3 | 4 | 3 | 0 | 0 | 0 | 0 | 0.31 |
| 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0.31 |

```
-------------------------------------------------------------------------------
|              13   |    10   |     0   |     0   |     0   |     0   |
-------------------------------------------------------------------------------

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
EL:   1 0.886E-02 0.848E-02-0.253E+08-0.151E+09-0.150E-09 0.986E+01 0.180E+02
EL:   2 0.166E-02-0.232E-04 0.494E+08-0.202E+09-0.100E-09-0.110E+01-0.510E+01
EL:   3 0.262E-02 0.692E-03 0.606E+08-0.203E+09-0.100E-09 0.362E+01 0.238E+01
EL:   4 0.899E-02 0.706E-02 0.606E+08-0.203E+09-0.150E-09 0.999E+01 0.151E+02
EL:   5 0.899E-02 0.894E-02-0.393E+08-0.799E+08-0.150E-09 0.999E+01 0.189E+02
EL:   6 0.353E-02 0.160E-02 0.606E+08-0.203E+09-0.100E-09 0.453E+01 0.420E+01
EL:   7 0.444E-02 0.251E-02 0.606E+08-0.203E+09-0.100E-09 0.544E+01 0.602E+01
EL:   8 0.535E-02 0.342E-02 0.606E+08-0.203E+09-0.100E-09 0.635E+01 0.784E+01
EL:   9 0.884E-02 0.782E-02 0.151E+08-0.190E+09-0.150E-09 0.984E+01 0.166E+02
EL:  10 0.626E-02 0.433E-02 0.606E+08-0.203E+09-0.100E-09 0.726E+01 0.966E+01
EL:  11 0.717E-02 0.524E-02 0.606E+08-0.203E+09-0.100E-09 0.817E+01 0.115E+02
EL:  12 0.808E-02 0.615E-02 0.606E+08-0.203E+09-0.100E-09 0.908E+01 0.133E+02
EL:  13 0.899E-02 0.706E-02 0.606E+08-0.203E+09-0.100E-09 0.999E+01 0.151E+02

    NR.  X-COORD   Y-COORD      VX         VY       CHARGE  X-WEIGHT  Y-WEIGHT
IO:   1 0.910E-03 0.910E-03 0.146E+06 0.146E+06 0.100E-09 0.191E+01 0.282E+01
IO:   2 0.182E-02 0.182E-02 0.146E+06 0.146E+06 0.100E-09 0.282E+01 0.464E+01
IO:   3 0.273E-02 0.273E-02 0.146E+06 0.146E+06 0.100E-09 0.373E+01 0.646E+01
IO:   4 0.364E-02 0.364E-02 0.146E+06 0.146E+06 0.100E-09 0.464E+01 0.828E+01
IO:   5 0.455E-02 0.455E-02 0.146E+06 0.146E+06 0.100E-09 0.555E+01 0.101E+02
IO:   6 0.546E-02 0.546E-02 0.146E+06 0.146E+06 0.100E-09 0.646E+01 0.119E+02
IO:   7 0.637E-02 0.637E-02 0.146E+06 0.146E+06 0.100E-09 0.737E+01 0.137E+02
IO:   8 0.727E-02 0.727E-02 0.146E+06 0.146E+06 0.100E-09 0.827E+01 0.155E+02
IO:   9 0.818E-02 0.818E-02 0.146E+06 0.146E+06 0.100E-09 0.918E+01 0.174E+02
IO:  10 0.909E-02 0.909E-02 0.146E+06 0.146E+06 0.100E-09 0.101E+02 0.192E+02
 ERROR CODE : 0
```