放射線医学総合研究所
801998016

HIMAC-018
NIRS-M-122

A Proton Dose Calculation Code for Treatment Planning Based on the Pencil Beam Algorithm

N. Kanematsu, T. Akagi, Y. Futami, A. Higashi, T. Kanai,

N. Matsufuji, H. Tomura and H. Yamashita

(Submitted to *Physics in Medicine and Biology*)

ペンシルビーム法陽子線量計算コード KDOSE マニュアル

兼松伸幸

1997 年 11 月

放射線医学総合研究所

〒263 千葉市稲毛区穴川 4-9-1

放射線医学総合研究所図書室
10. 8. 6

Submitted to Phys. Med. Biol.

A proton dose calculation code for treatment planning based on the pencil beam algorithm

Nobuyuki Kanematsu†, Takashi Akagi‡, Yasuyuki Futami§, Akio Higashi‡, Tatsuaki Kanai§, Naruhiro Matsufuji§, Hiromi Tomura§ and Haruo Yamashita§

† Communications Equipment Works, Mitsubishi Electric Corporation, 8-1-1 Tsukaguchi-Honmachi, Amagasaki, Hyogo 661, Japan
‡ Public Health Department, Hyogo Prefectural Government, 5-10-1 Shimoyamatedori, Chuo-Ku, Kobe, Hyogo 560, Japan
§ Division of Accelerator Physics and Engineering, National Institute of Radiological Sciences, 4-9-1 Anagawa, Inage-Ku, Chiba, Chiba 263, Japan

31 October 1997

Abstract. We have developed a 3D dose calculation code for proton treatment planning. In order to achieve a fast and accurate calculation in inhomogeneous matter like a human body, the code employs the pencil beam algorithm where a broad therapeutic beam is decomposed into finite number of narrow beams before entering the inhomogeneous material and the dose contributions from the individual beams are summed up to reconstruct the dose distribution. Comparisons with detailed Monte Carlo calculations showed good agreements in dose distributions while keeping the manageable computational time for treatment planning.

1. Introduction

A proton beam has a sharper and better depth-dose characteristics than conventional X rays and therefore it should be in particular effective for treatments where a very high precision is required. As the proton treatment technique has been improving in dose delivery precision, it is becoming of more importance to accurately calculate the delivered dose distribution in a patient. The dose calculation program for this purpose must handle the beam modifying devices in the treatment system and a patient. The functions of the devices should be appropriately modeled and the internal structure of a patient should be given by the planning CT images.

The theory of physical interactions of a proton with matter has been well established and therefore the Monte Carlo technique should be able to generally simulate therapeutic proton beams. It is, however, not practical to apply the Monte Carlo method to clinical proton treatment planning since it requires too heavy computations to handle even on a latest computer system. Two important beam properties involved in the dose calculation are the longitudinal depth-dose characteristics and the lateral beam spreading due to scattering. By assuming that the medium is laterally uniform, which is called the infinite slab layer approximation, the beam shape can be represented by a few parameters which can be semi-analytically calculated. However, this approximation is applicable only when the scattering

effect in inhomogeneous matter is negligible though that is not the case for a large field therapeutic beam in a patient. In other words, if the lateral beam size is small enough with respect to the inhomogeneity, the infinite slab layer approximation should be a fair approximation. By decomposing a broad therapeutic beam into subbeams narrow enough to apply the approximation before entering an inhomogeneous material, we can then calculate the dose contribution from each subbeam at any point to reconstruct the dose distribution, which is called the pencil beam algorithm. Since the scattering effect in a human body is typically a cm or less which is smaller than the typical organ size, the pencil beam algorithm is therefore a reasonable approach for a fast and accurate dose calculation required for treatment planning.

Efforts have been made to realize a fast and accurate 3D proton dose calculation based on the pencil beam algorithm by Petti (1992), Hong *et al.* (1996), and Carlsson *et al.* (1997), for example. Carlsson *et al.*'s work was focused on the behaviors of proton pencil beams in water to generate a basic data set to be used in treatment plan optimization for the beam scanning technique while the other two were intended to be used for treatment planning like this work is. A notable difference between the present work and Petti's or Hong *et al.*'s is found in algorithm as to how the pencil beams are generated and how the dose distribution is calculated at each of the given points of interest (POI's) in patient. In our "beam-oriented" algorithm, multiple subbeams are generated not with respect to POI's but rather independently with sufficient interval and number to cover the given irradiation field and then each subbeam is transported into the human body which is represented by a set of voxels given by the CT images. The dose contributions from a subbeam to the POI's are then calculated and added to reconstruct the dose distribution. On the other hand, in Petti's or Hong's "POI-oriented" algorithm, about 100 subbeams are generated around each POI to calculate the dose at that point and this procedure independently loops over all the POI's. The advantage of the beam-oriented algorithm is that it allows more detailed calculation in beam transport since the number of beams is minimal. The disadvantage is that each of the generated beams has to be associated with the POI's around the beam path out of the large number of POI's distributed in 3D space, which requires a smart algorithm to minimize the computations.

This work is intended to provide a reasonably fast and reasonably accurate dose calculation tool for clinical treatment planning for conventional irradiation systems, for which the wobbler-scatterer method is currently in our scope as a primary option for the beam spreading system. In this paper, we present the algorithm, the physical models, and the evaluated performance of the program in comparison with detailed Monte Carlo calculations.

2. Materials and Methods

2.1. Algorithm and modeling of the devices

The program employs a straightforward algorithm in such a way that a beam is tracked from upstream down toward the patient. The beam is defined as a bundle of particles and is characterized by beam direction, number of particles or fluence F, mean kinetic energy E, range straggling σ_R and three parameters to describe the lateral beam spread, namely, spatial variance V_{xx}, angular variance $V_{\theta\theta}$, and covariance $V_{x\theta}$, where the angular and the spatial spreads are approximated by Gaussian distribution functions. Those beam parameters are modified by physical interactions as later described in the text together with their definitions.

2.1.1. Range Modulator

A range modulating device such as a ridge filter is modeled as a variable thickness flat degrader. The modulation is simulated by superposing repeatedly calculated dose distributions with various degrader thickness and irradiation weight, which specify the device shape in the case of a ridge filter. The number of modulation steps may be minimized by considering the range straggling or by introducing some additional beam energy spread, in order to reduce calculation time.

2.1.2. Scatterers/degraders

The upper part of the irradiation system is assumed to be laterally uniform and the scatterers, the degraders, and air gaps in between are modeled as a 1-dimensional stack of materials through which we transport a single beam down until the beam meets the device which has some lateral structure. The scatterers/degraders have material parameters necessary to calculate physical interactions of the beam in matter, which are density ρ, average atomic-mass number ratio Z/A, mean excitation energy I, radiation length X_0, and nuclear interaction length at high energy λ_I. Parameters ρ, Z/A, and I for various materials are given by ICRU (1984) and X_0 and λ_I are given by Barnett *et al.* (1996) or are calculated according to the instructions given in the references.

2.1.3. Subbeam generation and wobbler system

The beam wobbling is approximated by a simple angular bending in a circular direction at a fixed height and is analytically included in the subbeam generation part of the code. That is, at the plane where lateral inhomogeneity begins, we decompose the parent beam into finite number of subbeams with infinitesimal spatial spread. The plane is divided into up to about 100×100 mesh-like pixels and we define a subbeam at center of each pixel. The subbeams directly inherit the properties of beam energy E and range straggling σ_R from the parent beam and the other subbeam parameters (marked by a prime) are analytically given by (Tomura 1997);

$$F'(r) = F\frac{\delta_x\delta_y}{2\pi V_{xx}}\exp\left(-\frac{r^2+r_W{}^2}{2V_{xx}}\right)I_0, \tag{1}$$

$$\theta'(r) = \frac{rV_{x\theta}}{V_{xx}} - \left(\frac{r_W V_{x\theta}}{V_{xx}} - \theta_W\right)\frac{I_1}{I_0}, \tag{2}$$

$$V_{\theta\theta}{}'(r) = V_{\theta\theta} - \frac{V_{x\theta}{}^2}{V_{xx}} + \left(\frac{r_W V_{x\theta}}{V_{xx}} - \theta_W\right)^2\left(\frac{1}{2} + \frac{I_2}{2I_0} - \frac{I_1{}^2}{I_0{}^2}\right), \tag{3}$$

where the parameters without a prime are ones for the parent beam, $I_k = \mathrm{I_k}(r\, r_W/V_{xx})$ is given by the k-th order modified Bessel function, $r = \sqrt{(x^2+y^2)}$ is the distance to the parent beam axis from the pixel, δ_x and δ_y are subbeam intervals, θ_W and r_W are the wobbling angle and radius at the plane, respectively, and θ' is the subbeam angle measured from the parent beam axis to determine the subbeam direction. We define the spatial variance V_{xx}' and the covariance $V_{x\theta}'$ for the subbeam as;

$$V_{xx}{}' = \frac{\delta_x{}^2 + \delta_y{}^2}{24}, \tag{4}$$

$$V_{x\theta}' = \frac{V_{x\theta}}{V_{xx}} V_{xx}', \tag{5}$$

where V_{xx}' is to give the spread corresponding to the uniform distribution in the pixel and $V_{x\theta}$ is to preserve the beam divergence due to scattering since $V_{xx}/V_{x\theta}$ corresponds to the mean distance to the scatterers distributed in the beam line. The subbeams are then transported independently through materials downward.

2.1.4. Compensator

Range compensators or boluses may be put in the beam line which are modeled as sets of mesh-like pixels with material thickness given to each pixel. For each of the subbeams, the compensator just behaves as a flat degrader with thickness defined at the pixel though which the beam passes, because of the infinite slab layer approximation.

2.1.5. Collimator

A beam collimator is defined by mesh-like pixels in a plane to represent the 2-dimensional collimator shape. A beam which hits a collimator pixel with "open" state will pass though the collimator and otherwise the beam is killed immediately. In this model, we ignore the thickness of the collimator and hence also the particles scattered by the edges of the collimator. Only the final collimator can be included while the other upstream ones are ignored. In a case where subbeams are generated before the collimator due to presence of a compensator upstream of the collimator, for example, a special corrective action to the beam spatial spread is applied. The parameter to give the beam size V_{xx}' is kept minimized until the beam comes down to the collimator plane. This is to enable the sharp cut-off at the collimator edges in order to correctly reproduce lateral penumbra, though the effect is very small in practice since the compensator is usually located very close to the collimator. In a usual case where the collimator is the first device with lateral structure, the subbeam pixels and the collimator pixels may be identically defined in the same plane.

2.1.6. Patient

A patient is modeled as a set of voxels of variable density water. The density of each voxel is derived from planning CT data so as to give the correct stopping power for therapeutic protons. The errors in scattering effect and nuclear effect due to this approximation are expected to be negligibly small in a typical treatment (Matsufuji 1997). Averaging of and/or interpolation from the CT data may be necessary to define voxels with the optimum size which may be a few mm in all three directions for a conventional large field irradiation. For simplicity of the code, a POI is defined at the center of each voxel, where dose value is calculated to construct 3D dose distribution. The dose value at a POI located at depth z along the beam and lateral distance r from the beam axis, $D(z,r)$, is then given by a simple Gaussian spreading of the linear energy transfer (LET) L which is the mean absorbed energy per unit length along the beam, as;

$$D(z,r) = L(z)\frac{1}{2\pi V_{xx}(z)}\exp\left(-\frac{r^2}{2V_{xx}(z)}\right). \tag{6}$$

The derivation of L is explained in the next subsection. Since we are not usually interested in fine structure of dose distribution within a voxel, we apply a limitation to V_{xx} by;

$$V_{xx} \geq \frac{w_x^2\left(1-u_x^2\right)+w_y^2\left(1-u_y^2\right)+w_z^2\left(1-u_z^2\right)}{24}, \tag{7}$$

where w_x, w_y, and w_z are the voxel widths and unit vector (u_x,u_y,u_z) represents the beam direction in the patient coordinate system. The right hand side term replaces V_{xx} when V_{xx} is smaller than that to calculate smeared dose instead of the point dose at the POI.

Since parameters L and V_{xx} are calculated sparsely at the transport steps, we apply the interpolation technique to obtain the value at the arbitrary depth. The dose at every POI is ideally calculated by summing up contributions from all the beams though it will be a very time-consuming computation to take all the combinations of the POI's and the beams. By ignoring the tail part of the Gaussian distribution beyond 3 standard deviations, we can restrict the region to look for the POI's to be associated to each beam and hence the combinations have been minimized. In the case of example 2 presented later in the text, which may be a typical example for treatment planning, about the same amount of CPU time was spent in the dose distributing part and in the beam transport part while the remaining overhead was negligible.

2.2. Modeling of physical interactions

2.2.1. Ionization energy loss

The mean energy loss rate per unit path length, or the stopping power S, for charged particles in matter due to ionization processes is given by the Bethe-Bloch formula (ICRU 1993). In a simplified form with such an approximation that the proton mass is much heavier than the electron mass and ignoring all the higher order corrections, we can write the formula as;

$$S = -\frac{dE}{dz} = 0.3071\rho\frac{Z}{A}\frac{1}{\beta^2}\left\{\ln\left(\frac{1.022\beta^2}{I\left(1-\beta^2\right)}\right)-\beta^2\right\}, \tag{8}$$

where β is the proton velocity in unit of speed of light. As to the units, MeV for energy, cm for length, and g/cm^3 for density are used unless noted, throughout this report. The higher order correction terms are usually very small and even much smaller than the error coming from the uncertainty of parameter I and hence are ignored. The proton range in water calculated by this formulation agrees with the fully corrected proton range table in the reference within 0.15 mm for incident kinetic energy everywhere up to 250 MeV. The relation between the proton kinetic energy E and the depth z is given by integrating this equation. In practice, the integration is done only once for water as the medium and values are stored in a table to associate kinetic energy E and water equivalent depth ζ which is defined by;

$$\zeta(z) = \int_0^z \frac{S(z')}{S_w(z')}dz', \tag{9}$$

where S_w is the stopping power for protons virtually passing through water with the given velocity. By numerically calculating ζ along the beam path, we can therefore obtain E at any depth by interpolation of the E–ζ table.

Energy spread due to fluctuation of the energy loss processes in thick material is well approximated by a Gaussian distribution (Fano 1964). Ignoring all the small higher order terms, the standard deviation of the Gaussian energy spread is given by;

$$\sigma_E{}^2 = \int 0.1569\rho \frac{Z}{A}\frac{1-\beta^2/2}{1-\beta^2}dz. \tag{10}$$

In practice, the integration is numerically performed step by step as depth and we convert the energy spread at the fixed position into the position spread in water equivalent depth at the fixed energy or so-called range straggling as;

$$\Delta\sigma_R{}^2 = \frac{\Delta\sigma_E{}^2}{S_w{}^2} = 0.1569\rho \frac{Z}{A}\frac{1-\beta^2/2}{1-\beta^2}\frac{\Delta z}{S_w{}^2}, \tag{11}$$

where prefix Δ represents the increase of the variable in the integration step.

The linear energy transfer from a proton L_p is then given by a convolution of the stopping power and the Gaussian function with standard deviation given by the range straggling, as;

$$L_p(z) = \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_R(z')}\exp\left(-\frac{\{\zeta(z)-\zeta(z')\}^2}{2\sigma_R(z')^2}\right)\frac{S(z')^2}{S_w(z')}dz'. \tag{12}$$

The histogram in figure 1 shows the derived L_p in water for one proton with 250 MeV incident kinetic energy while the open circles show the stopping power plotted at every mm before convolution of the straggling effect.

2.2.2. Inelastic Nuclear Interaction

A proton occasionally encounters the inelastic nuclear interaction with a nucleus in the medium, where a considerable part of the proton's kinetic energy is spent to destroy the nucleus and/or to generate secondary particles. In such cases, the charged secondary particles including the interacted proton are absorbed either immediately or in much shorter distance than the proton's original residual range. Neutrons and photons in the secondary particles usually escape far away from the interaction point and hence their total energy together with some spent for nuclear fission is seen as a missing energy. We included the effect of this phenomenon into LET calculation by a simple model.

For proton fluence reduction in water due to absorption, we employed the simplest fit to the experimental data by Lee *et al.* (1993), where the fluence linearly decreases as depth, and extended the model to other less important materials by scaling by nuclear interaction length at high energy. In a differential form, the proton fluence is given by;

$$\frac{dF}{d\zeta} = -F\frac{0.012}{1+0.012(R-\zeta)}\rho\frac{\lambda_{Iw}}{\lambda_I}, \tag{13}$$

where R is the proton range in water for the given incident energy, $R-\zeta$ is hence the residual range, and λ_{Iw} is the nuclear interaction length of water.

The dose contribution from secondary particles is formulated by the simplest model where a fixed energy escapes or is spent in the inelastic process and the rest of the energy is absorbed locally. The local energy deposition approximation is to assume that the proton and the charged secondary particles stop immediately after the interaction. The error from the approximation is expected to be large at material boundaries where density drastically changes such as body surface but relatively small in deeper region in body. In addition, the secondary particles should have less kinetic energy to move in the deeper region, which also justifies the approximation for therapeutic applications. We fitted the escaping energy to be

45 MeV to best reproduce the LET plots given in reference Carlsson *et al.* (1997) which were from model calculations based on the data from Seltzer (1993). The LET considering the inelastic nuclear interactions is then given by;

$$L = F \cdot L_p + \frac{-dF}{dz}(E - 45). \tag{14}$$

Figure 2 shows how the inelastic nuclear interaction is included in LET calculation in the code, where the solid, dashed, and dotted histograms represent the proton LET in water without the effect, one with the proton absorption effect, and one also with the contribution from the secondary particles, respectively, all normalized to one incident proton with kinetic energy of 250 MeV.

2.2.3. Multiple scattering

The multiple scattering of a charged particle in atomic Coulomb field is well represented by the Molière theory (Scott 1963) but the theory requires too complicated calculations to be used for this application. We therefore adopted the empirical approach to approximate the scattering angle by a Gaussian distribution with standard deviation given by the improved Highland formula, which was proposed by Lynch and Dahl (1991);

$$V_{\theta\theta}(z) = 13.6^2 \int_0^z \frac{dz'}{p^2\beta^2 X_0}\left(1 + 0.038 \ln \int_0^z \frac{dz'}{\beta^2 X_0}\right)^2, \tag{15}$$

where p is the proton momentum in MeV/c and $\sqrt{V_{\theta\theta}}$ corresponds to the standard deviation of the distribution of the particle angle with respect to the beam axis projected to one lateral axis. These integrations are numerically performed in practice, reflecting the proton energy changes to p and β. The beam transport step Δz is determined as the step which changes the kinetic energy relatively by 10% but no smaller than 0.1 mm, under such a constraint that the step should not exceed the boundary of different materials. In order to correct the small changes of p and β in each step, we made the following corrections in the stepwise calculations;

$$\frac{1}{\beta^2} \rightarrow \left(1 + \frac{S\Delta z}{2E}\right)\frac{1}{\beta^2}, \tag{16}$$

$$\frac{1}{p^2\beta^2} \rightarrow \left(1 + \frac{S\Delta z}{E}\right)\frac{1}{p^2\beta^2}, \tag{17}$$

where β and p in right hand sides are calculated at the beginning of each step by their definitions from the proton mass and the given kinetic energy. These correction formulae are derived from the approximations such that S is inversely proportional to β^2, that E is much smaller than the proton mass, and that $S\Delta z$ is much smaller than E. For example, the error in spatial spread of 250 MeV pencil beam in water due to the step size defined above amounted +0.3% compared to a very short step sized calculation, while, without the corrections, the error amounted –4.5%. Another small correction is applied to path length calculations since, due to angular spread, the average path length of the protons in a step is different from the step size which is the straight path along the beam. Except for the beam transport itself, the corrected step size is therefore used to all the calculations of the beam parameters, which is given by;

$$\Delta z \to (1+V_{\theta\theta})\Delta z\,. \tag{18}$$

This detour correction is derived from the average of the detour factor $1/\cos\theta$ with the small angle approximation, where the particle angle θ is given by the Gaussian distribution with standard deviation $\sqrt{(2V_{\theta\theta})}$. Since the detour correction itself is in fact very small for a proton beam, we ignored the radial dependence of the detour correction.

The angular deflections are then converted to the spatial displacement as the particle moves. The Fermi-Eyges theory extended by Brahme *et al.* (1981) gives how to transport the beam to obtain the spatial spread. Defining the mean scattering power T in the beam transport step Δz by;

$$T = \frac{\Delta V_{\theta\theta}}{\Delta z}\,, \tag{19}$$

the covariance $V_{x\theta}$ is given by the numerical integration of;

$$\Delta V_{x\theta} = V_{\theta\theta}\Delta z + \frac{T}{2}(\Delta z)^2\,, \tag{20}$$

and the spatial variance V_{xx} is similarly given by the numerical integration of;

$$\Delta V_{xx} = 2V_{x\theta}\Delta z + V_{\theta\theta}(\Delta z)^2 + \frac{T}{3}(\Delta z)^3\,. \tag{21}$$

2.3. Comparisons with Monte Carlo calculations

2.3.1. Monte Carlo program GEANT

We evaluated the performance of the code by comparing with Monte Carlo calculations where behaviors of individual particles are tracked down as precisely as possible and hence the systematic errors coming from the various approximations are minimized. For this purpose, we used GEANT version 3.21 for availability, reliability, and applicability to general particle simulations, which was originally developed for Particle Physics experiments (CERN 1994). Except for the hadron cutoff energy in water which we set to 1 MeV, we ran GEANT with options given by default, which means that the Bethe-Bloch formula with Landau's fluctuation theory for ionization loss, GHEISHA package (Fesefeldt 1985) for hadronic interactions, and the Molière theory for the multiple scattering, are used. However, a part of the GEANT code was modified for our purpose. Since GEANT lacks the capability to correctly implement chemical compounds for materials, water is usually defined by a mixture of hydrogen atoms and oxygen atoms. This treatment is causing an error to the stopping power by up to several percent since parameter I in the Bethe-Bloch formula is a chemical property. Therefore we defined water as a non-compound material and made a temporary hardwired modification to subroutine GDRELX to use user-defined value for parameter I for water to reproduce the range data by ICRU (1993) for 150 MeV proton.

In the following examples, we will demonstrate the performance of our code in terms of precision and speed in comparison with GEANT. The speed was measured by the elapsed time of PA8000 160 MHz CPU on Hewlett Packard C160 UNIX workstation which we currently plan to use for the near future medical facilities. It is noted that GEANT is not a program optimized for treatment planning and therefore the speed comparisons in this report do not quantitatively evaluate the Monte Carlo approach for treatment planning in general.

2.3.2. Example 1: a pencil beam in water

We first tested the code in the simplest case. A 150 MeV proton beam was incident into water and we calculated 3D dose distribution with POI's distributed at every 0.5 mm in x and y directions and every 1 mm in z direction in the pencil beam code. In GEANT, a volume of water was divided into 1 mm thick slices along the beam direction and the energy depositions from 10000 incident protons and secondary particles were accumulated in a 2-dimensional histogram by depth and by radius, utilizing the axial symmetry of the geometry. The longitudinal division of the volume was necessary to correctly count the deposited energy in the depth bins.

The computational time amounted much less than a second for the pencil beam calculation and about 10 minutes for GEANT. Figure 3 shows the LET normalized to one proton, reconstructed from the 3D dose distribution, where the solid and the dashed histograms are from the pencil beam and the GEANT calculations, respectively. The disagreement amounted typically a few percent of the peak LET and, at peak, GEANT gives smaller LET by several percent which was the maximum deviation. The isodose plots in z-x plane are presented in figure 4, where isodose lines are plotted at every 10% of the Bragg peak dose. These isodose plots look quite similar but are not able to show the tails below the 10% line. Figure 5 show the lateral dose profile at the depth of Bragg peak. Due to the limitation of the Gaussian approximation with the tail cut off in the pencil beam code, the disagreement in the tail region is in a relative sense large but is very small in absolute value.

2.3.3. Example 2: a spreaded beam in patient

We also tested the code for a typical treatment case, which means that a pencil beam is fed into the irradiation system, where the beam is spreaded and collimated, and applied to a patient. A 150 MeV proton beam was incident into the irradiation system, for which we put a wobbler system to give 25 mrad bending at 289 cm, a 0.67 mm thick lead scatterer at 252 cm, a 2 cm thick lucite degrader at 200 cm, a 10 cm diameter ring collimator at 22.5 cm, and a patient with a flat body surface located 10 cm below the collimator's bottom surface. The patient was modeled by a water phantom, in which we put a cylinder of double density water ($\rho = 2$ g/cm^3) in the region of depth in body 1 to 3 cm and radius 2 to 3 cm, as an example of inhomogeneity. In the pencil beam calculation, 64×64 subbeams with 1.5625 mm interval in two lateral directions were generated at the collimator plane and $\pi/4 \sim 80\%$ of them were transported into the 20×20×20 cm^3 patient volume consisting of 10^6 2×2×2 mm^3 voxels of water for each of which a POI was assigned at the center and the density was doubled if the POI was in the cylindrical region. On the other hand, in GEANT, one million of 150 MeV protons were fed into the irradiation system and the patient was modeled by water with a cylinder of double density water put in it, all divided along the beam axis at every 2 mm.

For the 3D dose calculation, the pencil beam code took 22 seconds while GEANT took about 1.5 hours. Figure 6 shows the isodose plots as functions of radius from the beam center and depth in body, where the hatched area corresponds to the double density cylindrical region. The overall shapes look quite similar though GEANT suffers from poor statistics at small radius. In order to see the details, we made slices of these plots along depth as in figure 7. The slices were made right on the double density region ($r = 2.5$ cm) for the top plot, at boundary ($r = 3$ cm) for the middle plot, and away from the inhomogeneity ($r = 4$ cm) for the bottom plot. Though the systematic dose difference at Bragg peak between the pencil beam calculation and GEANT still remains in these plots as in the previous example, it is shown in the top plot that the pencil beam code well reproduces the scattered component coming around the double density region, which made the small second peak in the plot.

3. Discussion

The substantial LET disagreement from GEANT at Bragg peak may be originated from the simplified treatment of physical interactions, especially, in inelastic nuclear interactions. However, GEANT has not been fully certified to be correct for this kind of application and hence we made no effort for the pencil beam code to fit with GEANT results. It is noted that the LET could be given by an external look-up-table based on the measurements, which includes all nuclear effects without modeling, instead of the internally calculated LET table as described in the text. For each subbeam in the pencil beam code, the lack of the tails of lateral beam spread causes some dose enhancement in the central region of the spread. That is however less significant in real therapeutic situations where many similar subbeams contribute to the dose at a POI. The pencil beam method is to be viewed as an approximation of the Monte Carlo method by bundling the particles into much less number of narrow beams. In expense of faster calculation speed, it lacks capability to correctly handle secondary particles and large angle scatterings especially at the collimator edges. Also, the infinite slab layer approximation in the pencil beam algorithm can produce large errors when there are large inhomogeneities within the spread of each individual beam, in particular, when the inhomogeneities are coherently integrated along the beam path. However, the infinite slab layer approximation is indispensable for our speed requirement and for simplicity of the code. If we were to minimize those errors, a fast Monte Carlo method on a parallel computer system would be the best approach.

In example 2, the pencil beam calculation took 22 seconds for a 150 MeV unmodulated proton beam of 10 cm diameter field in a patient with 10^6 POI's. With the usual range modulation of order of 10 cm, this code requires about 10 to 20 calculations with different degrader thickness, which will increase calculation time by the same factor. We therefore conclude that this code can handle a typical treatment plan with manageable time of several minutes. We also plan to include an option to utilize interpolation technique for range modulation simulation to reduce the number of modulation steps. Though we haven't quantitatively optimized the control parameters in the code such as voxel widths and subbeam intervals, the optimum numbers for them may be both about or somewhat smaller than the beam spread. It is, however, meaningless to have too small voxels compared to the beam spread due to the limitation of the infinite slab layer approximation. In order to achieve fast 3D dose calculation, it is essential to minimize the number of voxels and the number of subbeams while keeping the required resolution since, roughly speaking, the number of dose computations is proportional to the product of number of subbeams and the average number of POI's for a subbeam to interact with. So far, we have implicitly assumed the one-to-one correspondence between voxels and POI's for simplicity. However, it is also possible to define POI's independently on the voxels. If we reduce the number of POI's, the calculation will be accelerated by the reduction factor. It may be true that the POI-oriented algorithm can have smaller number of dose computations when the number of POI's is greatly reduced. However, in such a situation that we have to know the full dose distribution in patient, which we believe is natural for treatment planning, the beam-oriented algorithm must be simpler and faster. It is noted that the performance in terms of speed greatly depends on the way the code was written in practice rather than the principles on which the code is based.

Though the code currently supports only the wobbler-scatterer method as a beam spreading system, it won't be difficult to implement more options such as the dual-ring double scatterer method and various scanning methods when necessary. Inclusion of biological effects, when the basic data and the practical models are established, will be the task to be done in the future.

4. Conclusion

We have developed a reasonably fast and reasonably accurate 3D proton dose calculation code based on the pencil beam algorithm with the infinite slab layer approximation. The beam modifying devices and the internal structure of a patient are well handled by the code, which has been verified by comparisons with the Monte Carlo calculations. The code is basically ready to be installed into treatment planning systems as a practical proton dose calculation tool.

Acknowledgments

The authors are grateful to Drs. K. Kawachi, M. Endo, H. Koyama-Ito, S. Minohara, and N. Miyahara of National Institute of Radiological Sciences and Dr. A. Itano of Hyogo Prefectural Government for fruitful discussions and encouragement throughout this work.

References

Barnett R M *et al.* (Particle Data Group) 1996 Review of particle properties *Phys. Rev.* **D54** Part 1

Brahme A, Lax I and Andreo P 1981 Electron beam dose planning using discrete Gaussian beams. Mathematical background *Acta Radiol. Oncol.* **20** 147-158

Carlsson A K, Andreo P and Brahme A 1997 Monte Carlo and analytical calculation of proton pencil beams for computerized treatment plan optimization *Phys. Med. Biol.* **42** 1033-1053

CERN Application Software Group 1994 GEANT detector description and simulation tool *CERN Program Library Long Writeup* **W5013**

Fano U 1964 Penetration of protons, alpha particles, and mesons *Nuclear Science Series* **39** (*National Academy of Science – National Research Council Publication* **1133**) 287-352

Fesefeldt H C 1985 Simulation of hadronic showers, physics and applications *Technical Report PITHA* **85-02** III Physikalisches Institut, RWTH Aachen Physikzentrum, 5100 Aachen, Germany

Hong L, Goitein M, Bucciolini M, Comiskey R, Gottschalk B, Rosenthal S, Serago C and Urie M 1996 A pencil beam algorithm for proton dose calculations *Phys. Med. Biol.* **41** 1305-1330

ICRU 1984 Stopping powers for electrons and positrons *ICRU Report* **37**

ICRU 1993 Stopping powers and ranges for protons and alpha particles *ICRU Report* **49**

Lee M, Nahum A E and Webb S 1993 An empirical method to build up a model of proton dose distribution for a radiotherapy treatment-planning package *Phys. Med. Biol.* **38** 989-998

Lynch G R and Dahl O I 1991 Approximations to multiple Coulomb scattering *Nucl. Instrum. Methods* **B58** 6-10

Matsufuji N, Tomura H, Futami Y, Higashi A, Akagi T, Minohara S, Endo M and Kanai T 1997 Relationship between CT number and electron density, scatter angle and nuclear reaction for hadrontherapy treatment planning *To be published*

Petti P 1992 Differential-pencil-beam dose calculations for charged particles *Med. Phys.* **19** 137-149

Scott W T 1963 The theory of small-angle multiple scattering of fast charged particles *Rev. Mod. Phys.* **35** 231-313

Seltzer S M 1993 An assessment of the role of charged secondaries from nonelastic nuclear interactions by therapy proton beams in water *NIST Report* NISTIR 5221

Tomura H, Kanai T, Higashi A, Futami Y, Matsufuji N, Endo M, Soga F and Kawachi K 1997 Analysis of penumbra for uniform irradiation fields delivered by a wobbler method *To be published*

Figure Captions

Figure 1. LET from a 250 MeV proton in water. The stopping power for the mean kinetic energy at every mm is also plotted as open circles and the LET is shown as a histogram which includes the range straggling effect by Gaussian convolution.

Figure 2. LET normalized to one incident 250 MeV proton. The solid, the dashed, and the dotted histograms represent; without inelastic nuclear interactions, only with the proton absorption effect, and also with dose contribution from the secondary particles, respectively.

Figure 3. LET in water normalized to one incident 150 MeV proton, reconstructed from 3D dose distributions calculated by the pencil beam code (solid histogram) and by GEANT (dashed histogram).

Figure 4. Isodose plots in a cut-out plane. At left is shown the pencil beam calculation and at right is shown the GEANT calculation. The isodose lines are plotted at every 10% of the Bragg peak dose in each plot.

Figure 5. Lateral dose profile at Bragg peak in logarithmic scale. The solid and the dashed histograms show the pencil beam calculation and GEANT calculation, respectively.

Figure 6. Isodose plots in the modeled patient for the spreaded 150 MeV proton beam. At left is shown the pencil beam calculation and at right is shown the GEANT calculation. The hatched area in each plot represents the double density cylindrical region.

Figure 7. Longitudinal dose profile in the patient for the spreaded 150 MeV proton beam at three different radii; 2.5 cm (at top), 3.0 cm (at middle), and 4.0 cm (at bottom). The solid and dashed histograms represent the pencil beam calculation and GEANT calculation, respectively.

Figures

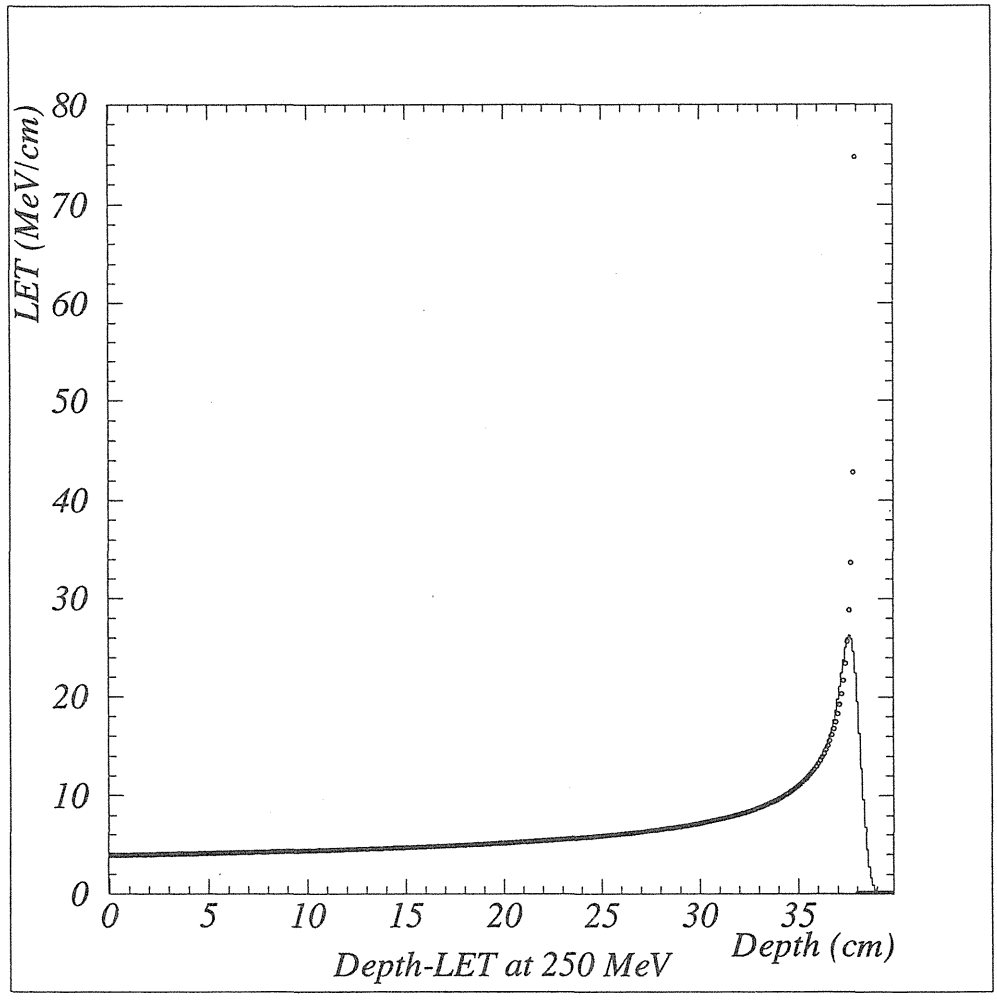

Figure 1. LET from a 250 MeV proton in water. The stopping power for the mean kinetic energy at every mm is also plotted as open circles and the LET is shown as a histogram which includes the range straggling effect by Gaussian convolution.

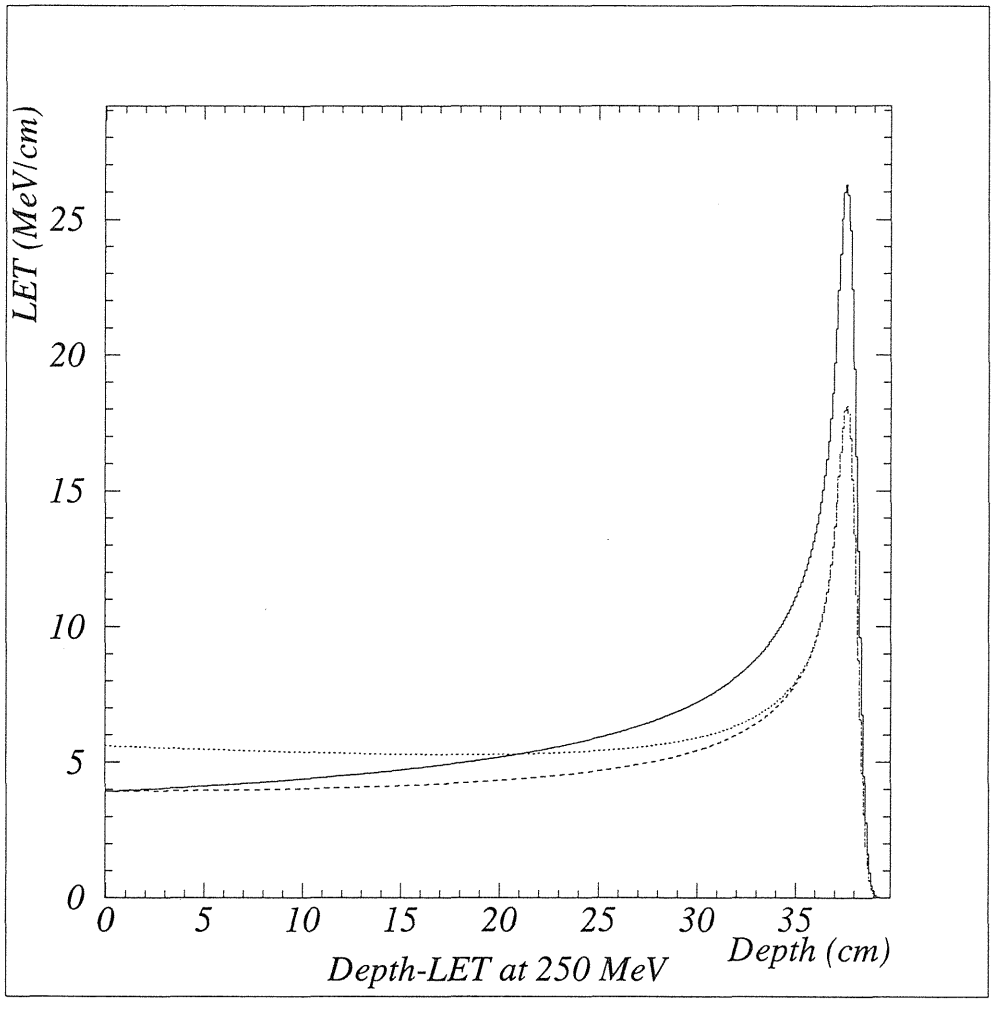

Figure 2. LET normalized to one incident 250 MeV proton. The solid, the dashed, and the dotted histograms represent; without inelastic nuclear interactions, only with the proton absorption effect, and also with dose contribution from the secondary particles, respectively.

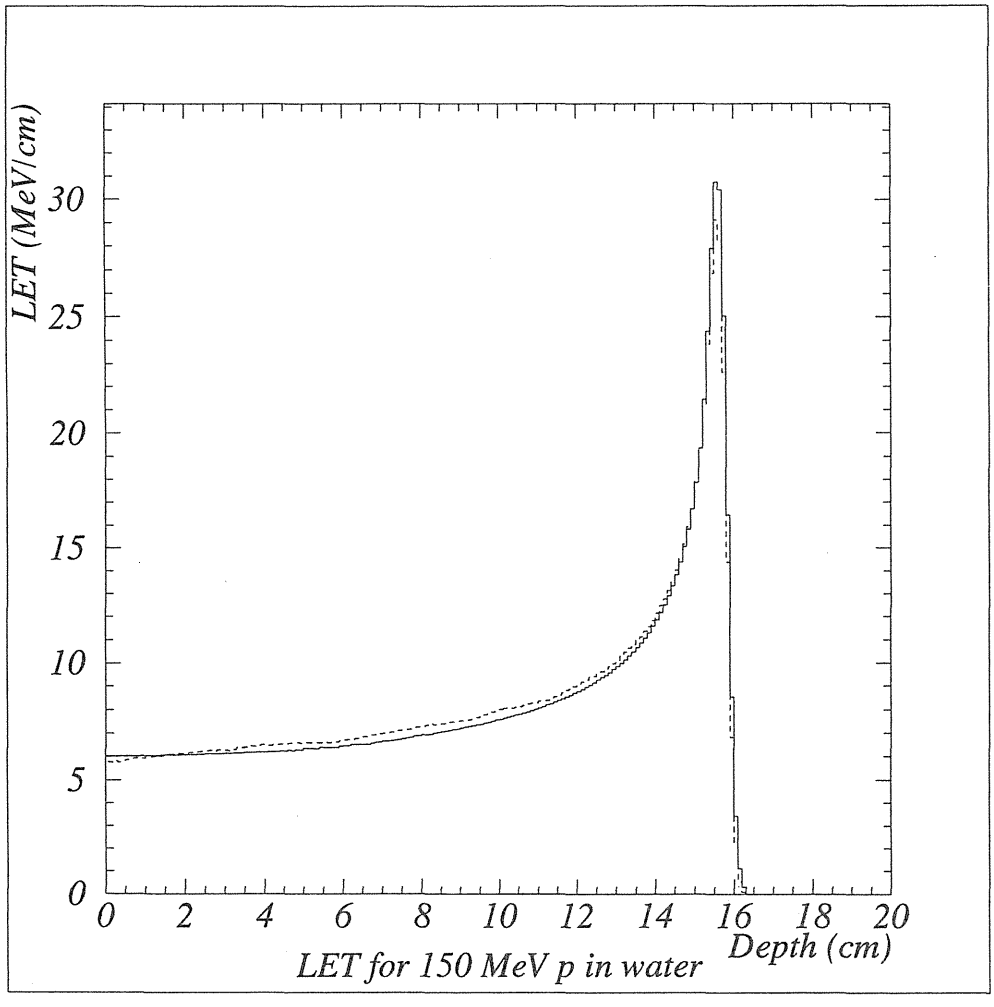

Figure 3. LET in water normalized to one incident 150 MeV proton, reconstructed from 3D dose distributions calculated by the pencil beam code (solid histogram) and by GEANT (dashed histogram).

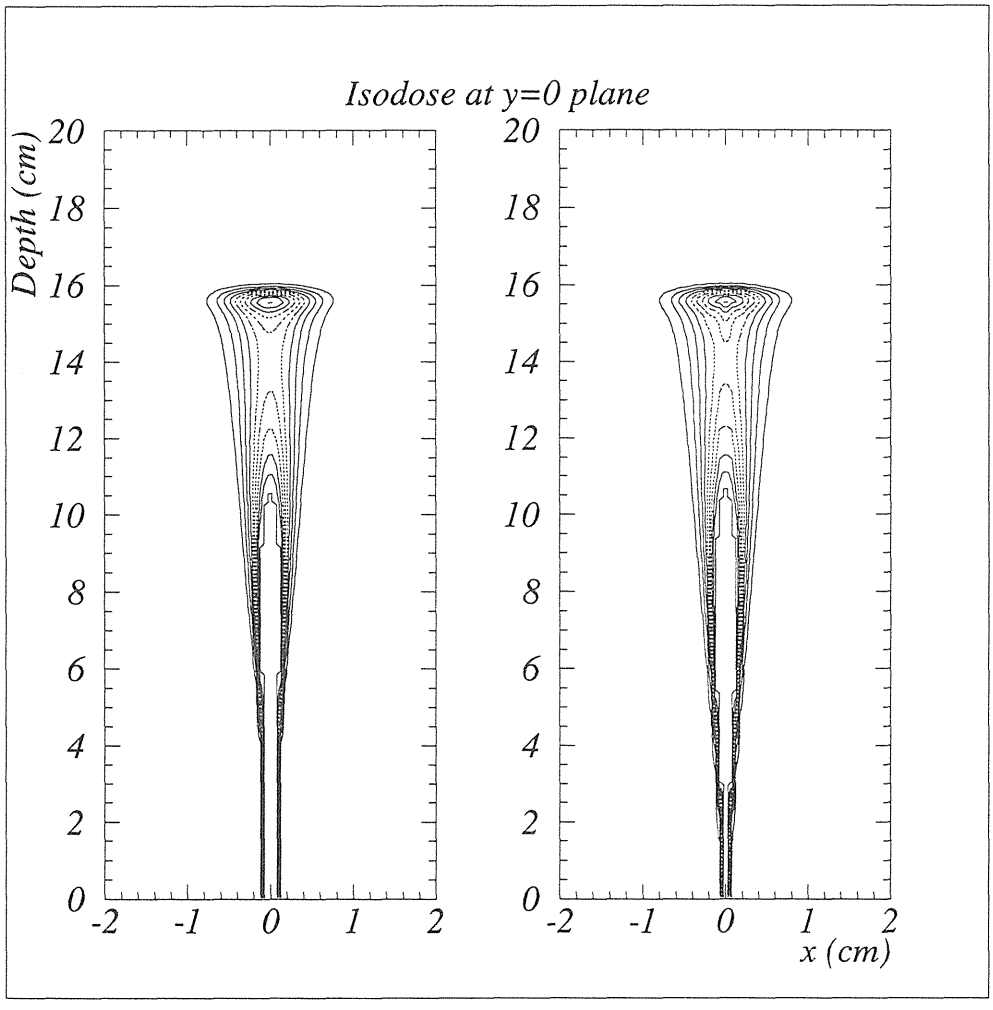

Figure 4. Isodose plots in a cut-out plane. At left is shown the pencil beam calculation and at right is shown the GEANT calculation. The isodose lines are plotted at every 10% of the Bragg peak dose in each plot.

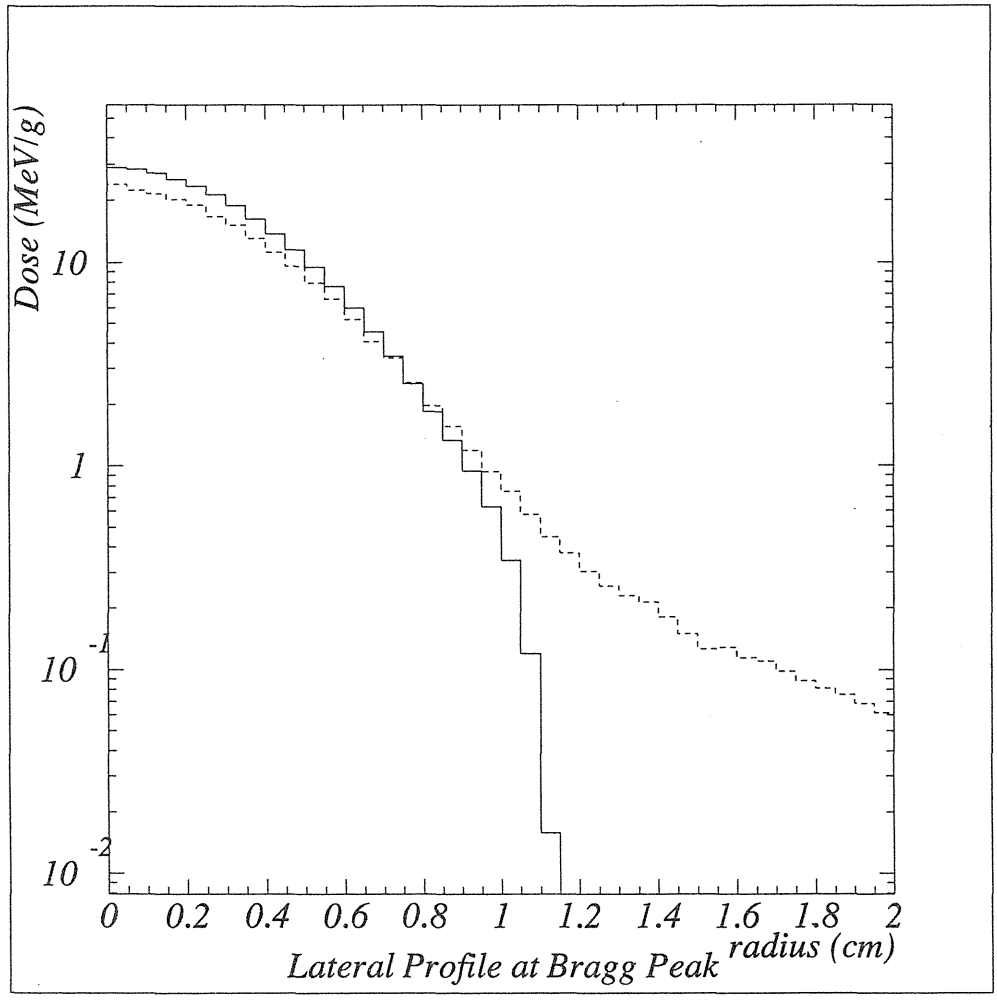

Figure 5. Lateral dose profile at Bragg peak in logarithmic scale. The solid and the dashed histograms show the pencil beam calculation and GEANT calculation, respectively.

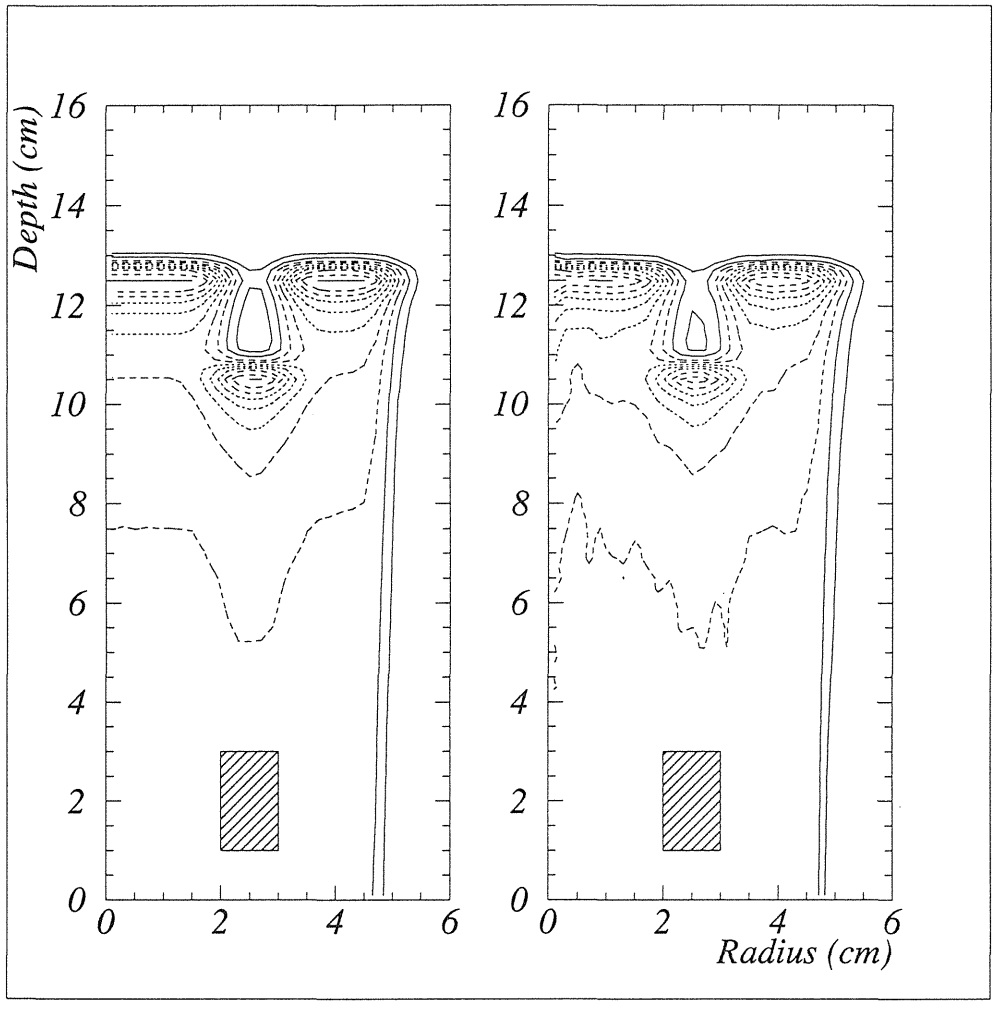

Figure 6. Isodose plots in the modeled patient for the spreaded 150 MeV proton beam. At left is shown the pencil beam calculation and at right is shown the GEANT calculation. The hatched area in each plot represents the double density cylindrical region.

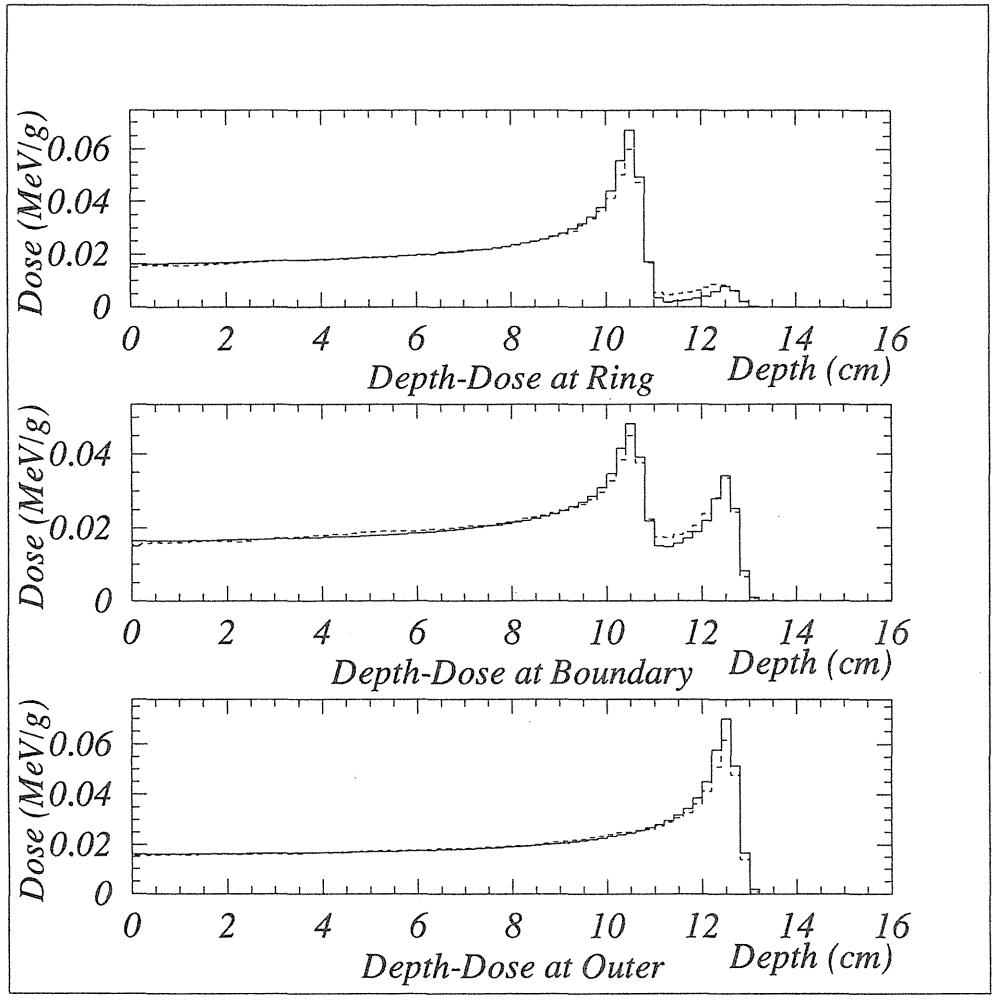

Figure 7. Longitudinal dose profile in the patient for the spreaded 150 MeV proton beam at three different radii; 2.5 cm (at top), 3.0 cm (at middle), and 4.0 cm (at bottom). The solid and dashed histograms represent the pencil beam calculation and GEANT calculation, respectively.

ペンシルビーム法陽子線量計算コード
KDOSE マニュアル

1997 年 10 月 30 日

三菱電機(株)通信機製作所 電子応用機器部 第一課 兼松 伸幸

(1997年2月～7月 放医研 重粒子治療センター 医用物理・工学研究部 研究生)

要旨

放医研・金井達明、外村浩美、松藤成弘、二見康之、山下晴男、兵庫県・赤城卓、東明男(敬称略)との共同研究において筆者が放医研研究生として開発した陽子線治療計画用ペンシルビーム法線量計算コードについてその内部を解説する。

1.概要

1. 1. 背景と目的

筆者は 1997 年 2 月より 7 月までの半年間放医研の研究生として重粒子治療センター医用重粒子物理・工学研究部第三研究室 金井達明室長らと共に陽子線治療計画用の線量計算コードの開発を行った。この計算コードは陽子線治療における治療計画作成のために人体内の線量計算を高速・高精度で行うことを目的とする。プログラムの名前は KDOSE とした。コード自身は 7 月中にほぼ完成し、概要および性能は日本医学物理学会の研究発表会(7 月 31 日茨城県つくば市)で発表し、ほぼ同じ内容で World Congress on Medical Physics and Biomedical Engineering(9 月 15 日フランス・ニース)においても発表した。また、同じ内容で英文の論文を作成中で、これは専門誌に投稿予定である。付録としてその原稿を添付する。このレポートは基本部分の完成した線量計算コードについてその内部構造を明らかにし、以後の応用や改良を容易にするためのものである。

1. 2. 使用言語

FORTRAN77 の標準に準拠したが、拡張仕様の内、INCLUDE 文、SAVE 文、DO...ENDDO 文、DOWHILE...ENDDO 文、インラインコメント(!)は使用した。変数名、関数名、サブルーチン名、コモンブロック名はすべて英 6 文字以内とした。すべての関数及びサブルーチンに"IMPLICIT NONE"宣言を課し、全変数を宣言文で定義した。

1. 3. 使用環境

上記の FORTRAN コードを正しくコンパイルできる FORTRAN コンパイラを必要とする。各社の UNIX ワークステーションならばすべて大丈夫と思われるが、これまで動作確認をしたのは HP(HPUX)、SGI(IRIX)、Sun(SunOS)である。線量計算コード自身は特殊なコンパイルオプションは必要としないが、誤差関数等の標準ライブラリは必要。HP では標準の誤差関数がなぜか正常に機能しなかったので CERN ライブラリ mathlib 中の誤差関数を使った。

1. 4. アルゴリズム

いわゆるペンシルビームアルゴリズムを採用した。ただし、MGH 等で開発されたコードとは異なり、モンテカルロ法の簡略化というアプローチをとり、照射野全体にペンシルビーム(サブビーム)を走らせて各ビームから各計算点への線量の寄与を加算していく方法を採用した。その他、コードで採用した物理的なモデルや性能などの説明は添付の論文内で説明されているのでここでは省略し、このレポートではプログラムの内部構造のみを説明する。

1. 5. 入出力

照射条件、患者データ等はコモンブロックの変数として計算に使われるが、現在のところ入出力に関しては FORTRAN のユーザールーチンを直接ユーザーが作成し、ここでコモン変数を操作することで行われる。従って、各ユーザーが各照射条件で実行ファイルを作成するか再コンパイルの必要のない入出力ルーチンを作成する必要がある。

1. 6. プログラムの流れ

メインプログラム KPMAIN で最初に呼ばれる UKINIT でユーザーは照射パラメータ、患者データ等を定義する。レンジモジュレーションはディグレーダの厚さを変えて複数回線量計算を行い、結果を加算することで実現する。ビームを輸送は 2 段階で行われ、まず 1 本のビームをサブビーム生成面

まで輸送し、ここでのビーム形状から生成面の各ピクセルでサブビームを定義し、それぞれを輸送していく。ビーム輸送は定義された物体(volume)ごとに KSTRAK により行われる。患者内での線量分布は各サブビームからの各計算点での線量寄与を加算することにより求める。すべての計算を終えたら最後に UKLAST が呼ばれ、ここでユーザーは結果の解析あるいは他プログラムへの受け渡しを行う。

1.7. 使い方例

次節で説明する全ファイルを一つのディレクトリに置き、UNIX コマンド make を実行することにより、実行ファイル kdose ができる。これを実行することにより、ukinit.f で定義される照射条件による線量計算結果は uklast.f で指定されるようにヒストグラムパッケージ HBOOK を使って二次元分布がとられ、HBOOK ファイル wobbler.hbook に格納される。これを paw 等を使って画面上に表示させることができる。HBOOK 及び PAW は CERN ライブラリの一部である。サンプルプログラムでは HBOOK を線量分布表示のために使用しているだけで、そうしない場合は KDOSE を動かすのに CERN ライブラリは不要である。

1.8. バグ等

このコードをリリースする時点では既知のバグは存在しないが、数例に対してのみしかテストされていないので広く使われれば改修しなければならないバグが発見されるかもしれない。バグ報告やこのコードをより良いものに改良するアイデア、あるいは実際に改造して効果があったこと等あれば、以下に一報して頂けると助かります。

郵便番号 661 兵庫県尼崎市塚口本町 8-1-1 三菱電機(株)通信機製作所
電子応用機器部 第一課 兼松 伸幸
電話 06-497-5026、FAX 06-497-5612、email kanemats@cew.melco.co.jp

2.ソースコードの詳細

2.1.メイクファイル

2.1.1.Makefile

SYSLIB = `cernlib mathlib packlib`	CERN ライブラリとリンク。HP では誤差関数を含む mathlib も必要であったが他では必要なし。
FFLAGS = -O -32 FFLAGS = -g -C -trapuv -32 FFLAGS = -g -C +ppu FFLAGS = +O2 +ppu -G	IRIX 用コンパイルオプション(CERN ライブラリとリンクするため 32 ビットモードにしている) 同、デバッグ用 HPUX デバッグ用コンパイルオプション 同、非デバッグ用 ハードウェアや使用目的に応じて適当にオプションを指定する。
INCLUDE = -I. F77 = f77	インクルードファイルの場所 FORTRAN コンパイラ名
all : kdose	
.SUFFIXES: .o .f	CERN ライブラリを使用するには、環境変数 CERN を CERN ライブラリを含むディレクトリツリーの大元のディレクトリ名で定義し、環境変数 CERN_LVL はそのディレクトリの下位に位置しリリースバージョンに相当するディレクトリ名で定義しておく。さらにコマンドパスに$CERN/$CERN_LVL/bin を加えておく。このメイクファイル内で使用しているコマンド cernlib はそこに置かれたシェルスクリプトである。 もしも UKLAST で HBOOK を使用しない場合は CERN ライブラリに関係するコンパイラオプション(IRIX では-32、HPUX では+ppu)は削ってもよい。IRIX で 64 ビットモードあるいは N32 モードでコンパイルすると速度が格段に向上するはず。
.f.o : $(F77) -c $(FFLAGS) $(INCLUDE) $<	
kdose : rklet.o rkint1.o rkint2.o ¥ dkaa.o kbmate.o kpmain.o ksptmt.o rkdxde.o rkintp.o ¥ dkbesi.o kbmodu.o ksptst.o rkekin.o rkresr.o ¥ rkenuc.o rksimp.o kbpati.o kstrak.o rkerf.o rkstop.o ¥ kbbeam.o kbphys.o kswobb.o rkestr.o ikpvox.o ikpxyz.o ¥ ksmear.o rkdfdr.o rkierf.o ¥ ksinit.o ukinit.o uklast.o ksptdo.o $(F77) $(FFLAGS) $(INCLUDE) -o kdose ¥ rklet.o rkint1.o rkint2.o ¥ dkaa.o kbmate.o kpmain.o ksptmt.o rkdxde.o rkintp.o ¥ dkbesi.o kbmodu.o ksptst.o rkekin.o rkresr.o ¥ rkenuc.o rksimp.o kbpati.o kstrak.o rkerf.o rkstop.o ¥	

```
        kbbeam.o kbphys.o kswobb.o rkestr.o ikpvox.o ikpxyz.o ¥
        ksmear.o rkdfdr.o rkierf.o ¥
        ksinit.o ukinit.o uklast.o ksptdo.o $(SYSLIB)
```

2.2. インクルードファイル

2.2.1. kcbeam.i

```
*---- KCBEAM - Initial Beam ------------------------------------------*
*                                                                     *
      REAL CHRGBM    ! charge [e]
      REAL AMASBM    ! mass [MeV/c^2]
      REAL FLUEBM    ! fluence (number of particles)
      REAL EKINBM    ! kinetic energy [MeV]
      REAL SEKBM     ! spread of kinetic energy [MeV]
      REAL XYZBM(3)  ! beam location
      REAL UVWBM(3)  ! beam direction
*                                                                     *
      COMMON/KCBEAM/CHRGBM, AMASBM, FLUEBM, EKINBM, SEKBM, XYZBM, UVWBM
*                                                                     *
*---- End of KCBEAM --------------------------------------------------*
```

照射系の受け取るビームの初期値を与えるためのコモンブロック

電荷(陽子は+1)
質量(同 938MeV)
粒子数
運動エネルギー
エネルギーの広がり(ガウシアン標準偏差)
ビーム入射位置
入射方向

KBBEAM あるいは UKINIT で定義される変数

2.2.2. kcbolu.i

```
*---- KCBOLU - Bolus -------------------------------------------------*
*                                                                     *
      INTEGER MBBO ! max number of boluses
      INTEGER MXBO ! max number of x division
      INTEGER MYBO ! max number of y division
      PARAMETER (MBBO=2, MXBO=256, MYBO=256)
*                                                                     *
      INTEGER IVOLBO(MBBO)          ! volume number
      INTEGER NXYBO(2, MBBO)        ! number of x,y divisions
      REAL XY1BO(2, MBBO)           ! lower left corner of first pixel [cm]
      REAL DXYBO(2, MBBO)           ! pixel size [cm]
      REAL HIGHBO(MXBO, MYBO, MBBO) ! signed height at each pixel [cm]
*                                                                     *
      COMMON/KCBOLU/IVOLBO, NXYBO, XY1BO, DXYBO, HIGHBO
```

ボーラスの形状データを与えるためのコモンブロック

最大 MBBO 個のボーラスが MXBO×MYBO の解像度で定義できる。

/KCVOLU/でこのボーラスに対応する物体の番号
ピクセル数
ボーラスの角の位置
ピクセルの大きさ
ボーラスの各ピクセルの厚さ

UKINIT で定義される変数

```
*                                                                    *
*---- End of KCBOLU -------------------------------------------------*
```

2.2.3. kccoll.i

```
*---- KCCOLL - Collimator -------------------------------------------*
*                                                                    *
      INTEGER MXCO       ! max number of x divisions
      INTEGER MYCO       ! max number of y divisions
      PARAMETER (MXCO=512,MYCO=512)
*                                                                    *
      INTEGER IVOLCO                 ! volume number
      INTEGER NXYCO(2)               ! number of x divisions
      REAL XY1CO(2)                  ! lower left corner of first pixel
      REAL DXYCO(2)                  ! pixel size
      LOGICAL OPENCO(MXCO,MYCO)      ! true if pixel is open
*                                                                    *
      COMMON/KCCOLL/IVOLCO,NXYCO,XY1CO,DXYCO,OPENCO
*                                                                    *
*---- End of KCCOLL -------------------------------------------------*
```

最終コリメータ形状を与えるためのコモンブロック

最大 MXCO×MYCO の解像度のピクセルで定義する。

/KCVOLU/でこのコリメータに対応する物体の番号
ピクセル数
コリメータの角の位置
ピクセルの大きさ
各ピクセルが開いているかどうか

UKINIT で定義される変数

2.2.4. kcdedo.i

```
*---- KCDEDO - depth-dose curve in water ----------------------------*
*                                                                    *
      INTEGER MDDD ! max number of data points
      PARAMETER (MDDD=256)
*                                                                    *
      REAL EKINDD        ! kinetic energy at depth=0.
      INTEGER NDDD       ! number of depth points
      REAL DEPTDD(MDDD)  ! depth in cmH2O
      REAL DOSEDD(MDDD)  ! measured LET in MeV/cm
*                                                                    *
      COMMON/KCDEDO/EKINDD,NDDD,DEPTDD,DOSEDD
*                                                                    *
*---- End of KCDEDO -------------------------------------------------*
```

実験値から深さ－線量特性を与えるためのコモンブロック

最大データ点数

深さ 0 でのビームの運動エネルギー
実際のデータ点数
深さ
LET

UKINIT で定義される変数

2.2.5. kcmate.i

```
*---- KCMATE - Material Data ------------------------------------------*
*                                                                      *
      INTEGER MAXMMT               ! max number of materials
      PARAMETER(MAXMMT=20)
*                                                                      *
      INTEGER IAIRMT               ! material number for air gaps
      INTEGER IWATMT               ! material number for water
      CHARACTER*16 NAMEMT(MAXMMT)  ! material name
      REAL ZOVAMT(MAXMMT)          ! effective Z/A
      REAL EXTEMT(MAXMMT)          ! mean excitation energy in MeV
      REAL DENSMT(MAXMMT)          ! density in g/cm^3
      REAL RADLMT(MAXMMT)          ! radiation length in g/cm^2
      REAL ABSLMT(MAXMMT)          ! nuclear absorption length in g/cm^2
*                                                                      *
      COMMON/KCMATE/IAIRMT, IWATMT, NAMEMT, ZOVAMT, EXTEMT, DENSMT, RADLMT
     &, ABSLMT
*                                                                      *
*---- End of KCMATE ---------------------------------------------------*
```

物質データを与えるためのコモンブロック

最大 MAXMMT 個の物質が定義できる。

空気に対応する物質番号
水に対応する物質番号
各物質の名前
Bethe-Bloch の式のパラメータ
同上
物質の密度
クーロン散乱計算のための輻射長
高エネルギーでの核反応の特性長

KBMATE あるいは UKINIT で定義される変数
物質データは ICRU37 (ICRU) 及び Review of Particle Properties (Particle Data Group) による。

2.2.6. kcmcon.i

```
*---- KCMCON - Mathematical Constants ---------------------------------*
*                                                                      *
      REAL*8 M_E         ! exp(1)
      REAL*8 M_LOG2E     ! log2(e)
      REAL*8 M_LOG10E    ! log10(e)
      REAL*8 M_LN2       ! ln(2)
      REAL*8 M_LN10      ! ln(10)
      REAL*8 M_PI        ! pi
      REAL*8 M_SQRT2     ! sqrt(2)
      REAL*8 M_SQRTPI    ! sqrt(pi)
*                                                                      *
      PARAMETER (M_E         = 2.7182818284590452354 )
      PARAMETER (M_LOG2E     = 1.4426950408889634074 )
      PARAMETER (M_LOG10E    = 0.43429448190325182765)
```

数学的な定数の定義

```
      PARAMETER (M_LN2       = 0.69314718055994530942)
      PARAMETER (M_LN10      = 2.30258509299404568402)
      PARAMETER (M_PI        = 3.14159265358979323846)
      PARAMETER (M_SQRT2     = 1.41421356237309504880)
      PARAMETER (M_SQRTPI    = 1.77245385090551602729)
*                                                               *
*---- End of KCMCON ------------------------------------------*
```

2.2.7. kcmodu.i

```
*---- KCMODU - Range Modulator -------------------------------*
*                                                               *
      INTEGER MSTPMD       ! max number of steps
      PARAMETER (MSTPMD=64)
*                                                               *
      INTEGER IVOLMD       ! volume number
      INTEGER NSTPMD       ! number of steps
      REAL THICMD(MSTPMD)  ! effective thickness of each step
      REAL RATIMD(MSTPMD)  ! intensity at each step
*                                                               *
      COMMON/KCMODU/IVOLMD,NSTPMD,THICMD,RATIMD
*                                                               *
*---- End of KCMODU ------------------------------------------*
```

レンジモジュレーションの仕方を決めるコモンブロック

最大 MSTPMD ステップの飛程調整

/KCVOLU/でこのレンジモジュレータに対応する物体番号
モジュレーションステップ数
各ステップの飛程シフト量
各ステップでの照射粒子数

KBMODU あるいは UKINIT で定義される変数

2.2.8. kcpati.i

```
*---- KCPATI - Patient ---------------------------------------*
*                                                               *
      INTEGER MXYZPT          ! max number of voxels
      PARAMETER (MXYZPT=128*128*128)
*                                                               *
      INTEGER IVOLPT          ! volume number
      REAL XYZOPT(3)          ! local origin in VL system
      REAL RP2VPT(3,3)        ! rotation matrix local to VL
      REAL RV2PPT(3,3)        ! rotation matrix VL to local
      INTEGER NXYZPT(3)       ! number of x,y,z divisions
      REAL XYZ1PT(3)          ! outer corner of first voxel [cm]
```

計画用 CT 画像をベースにした患者データのためのコモンブロック

三次元計算ボクセルの最大数

/KCVOLU/で患者に対応する物体番号
患者座標系の原点のビーム座標系(KCVOLU)での位置
患者座標系からビーム座標系への回転行列
ビーム座標系から患者座標系への回転行列
ボクセル数
ボクセルの角

`      REAL DXYZPT(3)          ! (dx,dy,dz) of voxel increment [cm]`	ボクセル幅
`      REAL DENSPT(MXYZPT)     ! water equivalent density [gH2O/cm^3]`	各ボクセルの水等価密度
`      REAL DOSEPT(MXYZPT)     ! physical dose [MeV/gH2O]`	各ボクセル中心での線量値
`*                                                                *`	
`      COMMON/KCPAT1/IVOLPT,XYZOPT,RP2VPT,RV2PPT,NXYZPT,XYZ1PT,DXYZPT`	KBPATI あるいは UKINIT で定義される変数
`      COMMON/KCPAT2/DENSPT`	同上
`      COMMON/KCPAT3/DOSEPT`	UKLAST で処理するための計算結果
`*                                                                *`	
`*---- End of KCPATI ---------------------------------------------*`	

2.2.9. kcpcon.i

`*---- KCPCON - Physical Constants -------------------------------*`	物理定数の定義
`*                                                                *`	
`*     Default Units:                                             *`	基本単位は
`*       time     - sec                                           *`	時間は秒、
`*       length   - cm or g/cm^2 or gH2O/cm^2                     *`	長さはセンチメートルあるいはグラム毎平方センチメートルあるいは水等価厚
`*       angle    - radian                                        *`	角度はラジアン、
`*       charge   - positron charge                               *`	電荷は陽電子電荷量、
`*       energy   - MeV                                           *`	エネルギーは MeV、
`*       momentum - MeV/c                                         *`	運動量は MeV/c
`*       mass     - MeV/c^2 or g                                  *`	質量は MeV/c^2(粒子)あるいはグラム(物体)
`*       density  - g/cm^3                                        *`	密度はグラム毎立方センチメートル
`*                                                                *`	
`      REAL*8 COULOM ! e to coulomb conversion factor`	
`      REAL*8 GRAM   ! MeV/c^2 to g conversion factor`	
`      REAL*8 JOULE  ! MeV to Joule Conversion Factor`	
`      REAL*8 C      ! speed of light in cm/sec`	
`      REAL*8 ALPHA  ! fine structure constant`	
`      REAL*8 EMASS  ! electron mass in MeV/c^2`	
`      REAL*8 ERADI  ! classical electron radius in cm`	
`      REAL*8 AVOGAD ! Avogadro's Number (number of particles per mol)`	
`      REAL*8 AMU    ! atomic mass unit in MeV/c^2`	
`      REAL*8 PMASS  ! proton mass in MeV/c^2`	
`*                                                                *`	
`      PARAMETER (COULOM=1.602 177 33 D-19)`	

```
      PARAMETER (GRAM=1.782 662 70 D-39)
      PARAMETER (JOULE=1.602 177 33 D-13)
      PARAMETER (C=2.997 924 58 D10)
      PARAMETER (ALPHA=1.D0/137.035 989 5 D0)
      PARAMETER (EMASS=0.510 999 06 D0)
      PARAMETER (ERADI=2.817 940 92 D-13)
      PARAMETER (AVOGAD=6.022 136 7 D23)
      PARAMETER (AMU=931.494 32 D0)
      PARAMETER (PMASS=938.272 31 D0)
*                                                                     *
*---- End of KCPCON ------------------------------------------------*
```

2.2.10. kcpenc.i

```
*---- KCPENC - Subbeam Parameters ----------------------------------*
*                                                                     *
      INTEGER IVOLPB             ! volume number for dummy volume
      INTEGER NXYPB(2)           ! number of pixels
      REAL XY1PB(2)              ! corner of the first pixel
      REAL DXYPB(2)              ! pixel size
*                                                                     *
      COMMON/KCPENC/IVOLPB,NXYPB,XY1PB,DXYPB
*                                                                     *
      INTEGER MPPB   ! number of parameters
      PARAMETER (MPPB=20)
*                                                                     *
      INTEGER KCHGPB ! charge [e]
      INTEGER KMASPB ! mass [MeV/c^2]
      INTEGER KXOPB  ! x of beam origin [cm]
      INTEGER KYOPB  ! y of beam origin [cm]
      INTEGER KZOPB  ! z of beam origin [cm]
      INTEGER KDXPB  ! x component of beam direction
      INTEGER KDYPB  ! y component of beam direction
      INTEGER KDZPB  ! z component of beam direction
      INTEGER KDISPB ! distance from origin
      INTEGER KFLUPB ! fluence (number of particles)
```

サブビーム生成のためのコモンブロック

生成平面に対する/KCVOLU/での物体番号
生成平面のピクセル数
ピクセルの角の位置
ピクセルの大きさ

KBPATI あるいは UKINIT で定義される変数

ビームパラメータの最大数

電荷(を与えるインデックス番号)
質量
ビーム原点位置

ビーム方向

原点から現在の輸送点までの距離
粒子数

`      INTEGER KEKIPB ! kinetic energy [MeV]`	運動エネルギー
`      INTEGER KDEPPB ! integrated water equivalent depth`	積分水等価深さ
`      INTEGER KX1PB  ! integrated radiation length 1`	積分散乱パラメータ1
`      INTEGER KX2PB  ! integrated radiation length 2`	積分散乱パラメータ2
`      INTEGER KAAPB  ! mean projected scattering angle squared [rad^2]`	横方向の1軸に射影した散乱角の分散
`      INTEGER KTTPB  ! mean projected displacement squared [cm^2]`	横方向の1軸上の散乱変位の分散
`      INTEGER KATPB  ! covariance [rad cm]`	相関項
`      INTEGER KRRPB  ! range straggling squared [(gH2O/cm^2)^2]`	レンジストラグリングの二乗
`*                                                                *`	
`      PARAMETER (KXOPB=3,KYOPB=4,KZOPB=5,KDXPB=6,KDYPB=7,KDZPB=8`	各ビームパラメータのインデックス番号
`     &,KCHGPB=1,KMASPB=2,KFLUPB=9,KEKIPB=10,KDISPB=11,KAAPB=12`	
`     &,KATPB=13,KTTPB=14,KRRPB=15,KDEPPB=16,KX1PB=17,KX2PB=18)`	
`*                                                                *`	
`*---- End of KCPENC ---------------------------------------------*`	

2.2.11. kcphys.i

`*---- KCPHYS - Physics options ------------------------------------*`	物理モデルのオプション
`*                                                                *`	
`      INTEGER ISCAPH ! scattering`	散乱のモデル
`                     ! 1:Highland`	1:ハイランドの式
`                     ! 2:Lynch-Dahl`	2:リンチ-ダールの式
`      INTEGER ILETPH ! LET calculation`	LET 計算法
`                     ! 1:theoretical step by step calculation`	1:ステップ毎の理論計算
`                     ! 2:theoretical depth-dose data`	2:テーブル化した理論計算
`                     ! 3:experimental depth-dose`	3:実験値テーブル/KCDED0/の利用
`      INTEGER ISTRPH ! range straggling`	レンジストラグリング
`                     ! 0:no additional straggling`	0:なし(ILETPH=3 の時)
`                     ! 1:gaussian theory`	1:ガウシアン理論計算
`                     ! 2:1.1% of range`	2:水等価深さの 1.1%
`      INTEGER INUCPH ! nuclear interaction 0:off, 1:on`	核反応の効果(0:なし、1:あり)
`      REAL    ECUTPH ! cut off energy [MeV]`	陽子のカットオフ運動エネルギー
`      REAL    RELKPH ! max of relative kinetic energy change`	ステップあたり運動エネルギー変化率の最大値
`      REAL    STMIPH ! minimum step size [cm]`	最小ステップサイズ
`      REAL    STMAPH ! maximum step size [cm]`	最大ステップサイズ
`      REAL    SIGRPH ! range in sigma to define longitudinal spread`	線量計算領域を決めるための、標準偏差を単位としたピーク深さからの最大深さ

`      REAL    SIGTPH ! range in sigma to define lateral spread`	同、横方向の最大半径
`      REAL    DSIZPH ! dosimeter size (negative -> voxel size) [cm]`	線量計算の仮想線量計の大きさ(負ならばボクセルサイズ)
`*                                                                  *`	
`      COMMON/KCPHYS/ISCAPH, ILOSPH, ISTRPH, INUCPH`	KBPHYS あるいは UKINIT で定義される変数
`     &, ECUTPH, RELKPH, STMIPH, STMAPH, SIGRPH, SIGTPH, DSIZPH`	
`*                                                                  *`	
`*---- End of KCPHYS ----------------------------------------------*`	

2. 2. 12. kcvolu. i

`*---- KCVOLU - 1-Dimensional Volume Definitions --------------------*`	ビームライン上の物体を定義するためのコモンブロック
`*                                                                  *`	
`*     Coordinate system in this common                             *`	ここでの座標系(ビーム座標系)はアイソセンターを原点に、入射ビームの反対の方向をz方向とする。
`*        Origin      = isocenter                                   *`	
`*        Z direction = -beam direction                             *`	
`*                                                                  *`	
`      INTEGER MAXVVL             ! max number of volumes`	定義できる物体の最大数
`      PARAMETER (MAXVVL=20)`	
`*                                                                  *`	
`      REAL RV2GVL(3,3)           ! rotation matrix from VL to global`	ビーム座標系から治療室座標系への変換するための回転行列
`      REAL RG2VVL(3,3)           ! inverse matrix`	逆行列
`      INTEGER NVOLVL             ! number of defined volumes`	定義された物体の数
`      CHARACTER*4 NAMEVL(MAXVVL) ! volume name`	各物体の名前
`      CHARACTER*4 CTYPVL(MAXVVL) ! volume type`	各物体の種類コード
`      INTEGER ISUBVL(MAXVVL)     ! index in volume-type specific common`	サブインデックス(種類による)
`      INTEGER IMATVL(MAXVVL)     ! material number`	材質番号
`      REAL ZPOSVL(MAXVVL)        ! volume center height [cm]`	物体の中心位置の高さ
`                                 ! flat surface height [cm] for bolus`	ボーラスの場合はフラット面の高さ
`      REAL THICVL(MAXVVL)        ! volume thickness [cm]`	物体の厚さ
`      REAL PHIVL(MAXVVL)         ! rotation angle about z [rad]`	ビーム軸回りの回転角
`*                                                                  *`	
`      COMMON/KCVOLU/RV2GVL, RG2VVL, NVOLVL, NAMEVL`	UKINIT で定義される変数
`     &, CTYPVL, ISUBVL, IMATVL, ZPOSVL, THICVL, PHIVL`	
`*                                                                  *`	
`*     Volume type definitions                                      *`	種類コード名
`      CHARACTER*4 CWOBVL         ! wobbler system`	ワブラー

```
      CHARACTER*4 CPENVL          ! subbeam generation
      CHARACTER*4 CDEGVL          ! degrader
      CHARACTER*4 CMODVL          ! range modulator
      CHARACTER*4 CCOLVL          ! collimator
      CHARACTER*4 CBOLVL          ! bolus
      CHARACTER*4 CPATVL          ! patient
      PARAMETER (CWOBVL='WOBB',CPENVL='PENC',CDEGVL='DEGR',CMODVL='MODU'
     &,CCOLVL='COLL',CBOLVL='BOLU',CPATVL='PATI')
*                                                                     *
*---- End of KCVOLU ------------------------------------------------*
```

サブビーム生成面
ディグレーダ
レンジモジュレータ
コリメータ
ボーラス
患者

2.2.13. kcwobb.i

```
*---- KCWOBB - Wobbler System --------------------------------------*
*     Simplest approximation of circular wobbler system
*                                                                     *
      INTEGER IVOLWB ! volume number for wobbler system
      REAL ANGLWB    ! wobbling angle [rad]
*                                                                     *
      COMMON/KCWOBB/IVOLWB,ANGLWB
*                                                                     *
*---- End of KCWOBB ------------------------------------------------*
```

ワブラー条件を与えるコモンブロック
磁石の厚さ、X, Y方向の位置の差を無視

/KCVOLU/でワブラーに対応する物体の番号
曲げる角度

UKINIT で定義される変数

2.3. ブロックデータ

2.3.1. kbbeam.f

```
************************************************************************
      BLOCK DATA KBBEAM ! block data for /KCBEAM/
*                                                                     *
      IMPLICIT NONE
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
************************************************************************

      INCLUDE 'kcpcon.i'
      INCLUDE 'kcbeam.i'
```

/KCBEAM/に初期値を与えるブロックデータ

```
      DATA CHRGBM /1./
      DATA AMASBM /PMASS/
      DATA FLUEBM /1./
      DATA EKINBM /230./
      DATA SEKBM  /0./
      DATA XYZBM  /3*0./
      DATA UVWBM  /3*0./

      END
```

2.3.2. kbmate.f

```
************************************************************************
      BLOCK DATA KBMATE ! block data for /KCMATE/
*                                                                      *
      IMPLICIT NONE
*                                                                      *
*     Default material definitions                                     *
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcmate.i'

      DATA IAIRMT /1/
      DATA IWATMT /2/
      DATA NAMEMT /'AIR         ','WATER       '
     &            ,'LUCITE      ','POLYETHYLEN '
     &            ,'ALUMINUM    ','IRON        '
     &            ,'COPPER      ','BRASS       '
     &            ,'TUNGSTEN    ','LEAD        '
     &            ,10*'            '/

      DATA ZOVAMT /0.4998, 0.5551, 0.5394, 0.5703, 0.4818,
     &             0.4656, 0.4564, 0.4552, 0.4025, 0.3958,
     &             10*0./
```

/KCMATE/に初期値を与えるブロックデータ

とりあえずよく使いそうな 10 個だけ定義されている。

```
      DATA EXTEMT /85.7E-6,   75.0E-6,   74.0E-6,   57.4E-6,  166.4E-6,
     &             286.0E-6,  321.9E-6,  333.6E-6,  725.2E-6,  820.0E-6,
     &             10*0./

      DATA DENSMT /0.0012,  1.0000,  1.1900,  0.9400,  2.7000,
     &              7.8700,  8.9600,  8.4890,  19.300,  11.350,
     &              10*0./

      DATA RADLMT /36.66,  36.08,  40.55,  44.78,  24.01,
     &             13.84,  12.86,  12.30,   6.76,   6.37,
     &             10*0./

      DATA ABSLMT / 90.0,   84.9,   83.6,   78.8,   106.4,
     &             131.9,  134.9,  141.70,  185.0,  194.0,
     &             10*0./

      END
```


2.3.3. kbmodu.f

```
************************************************************************
      BLOCK DATA KBMODU ! block data for /KCMODU/
*                                                                      *
      IMPLICIT NONE
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcmodu.i'

      DATA IVOLMD/0/
      DATA NSTPMD/1/
      DATA THICMD/MSTPMD*0./
      DATA RATIMD/MSTPMD*1./
      END
```

/KCMODU/に初期値を与えるブロックデータ

レンジモジュレーションなし。

2.3.4. kbpati.f

```
************************************************************************
      BLOCK DATA KBPATI ! block data for /KCPATn/
*                                                                      *
      IMPLICIT NONE
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcpati.i'

      DATA IVOLPT/0/
      DATA XYZOPT/0.,0.,0./
      DATA RP2VPT/1.,0.,0.,0.,1.,0.,0.,0.,1./
      DATA RV2PPT/1.,0.,0.,0.,1.,0.,0.,0.,1./
      DATA NXYZPT/3*1/
      DATA XYZ1PT/3*0./
      DATA DXYZPT/3*1./
      DATA DENSPT/MXYZPT*1./
      DATA DOSEPT/MXYZPT*0./
      END
```

/KCPAT1/, /KCPAT2/, /KCPAT3/に初期値を与えるブロックデータ

2.3.5. kbphys.f

```
************************************************************************
      BLOCK DATA KBPHYS ! block data for /KBPHYS/
*                                                                      *
      IMPLICIT NONE
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcphys.i'

      DATA ISCAPH /2/
      DATA ILETPH /1/
```

/KBPHYS/に初期値を与えるブロックデータ

散乱は Lynch-Dahl の式
LET は各ビームについて計算

DATA ISTRPH /1/ DATA INUCPH /1/ DATA ECUTPH /1.0/ DATA RELKPH /0.1/ DATA STMIPH /0.1/ DATA STMAPH /5.0/ DATA SIGRPH /3.0/ DATA SIGTPH /3.0/ DATA DSIZPH /-1.0/ END	レンジストラグリングはガウシアン理論 核反応の効果を含む 1MeV まで輸送する 10%までの運動エネルギー変化を許す 最小ステップサイズ 100 ミクロン 最大ステップサイズ 5cm 深さ方向にはピークから 3 シグマまでとる 横方向にはビーム中心から 3 シグマとる 線量計のサイズはボクセルサイズ

2.4. メインプログラム

2.4.1. kpmain.f

** PROGRAM KPMAIN ! the main program for wobbler method * * IMPLICIT NONE * * * 19970725 - N. Kanematsu (Mitsubishi Electric Corporation) * ** INCLUDE 'kcwobb.i' ! wobbler INCLUDE 'kcbeam.i' ! beam INCLUDE 'kcvolu.i' ! beamline devices INCLUDE 'kcmate.i' ! material data INCLUDE 'kcmodu.i' ! range modulation INCLUDE 'kcpenc.i' ! subbeam generation INCLUDE 'kccoll.i' ! collimator REAL RKSTOP INTEGER IU, IV, IVOL, NVOL, ISTP REAL PARM1(MPPB), PARM2(MPPB) REAL ZPOS, PHI, SPHI, CPHI, U, V, R, FLUX, ANGL, SIGM, BESQ, CHSQ, ZW, AW, DW REAL AA, AT, TT	メインプログラム

```
      LOGICAL LSPR

C     User setup routine
      CALL UKINIT

C     Initialize look-up tables
      CALL KSINIT

C     Repeat number of range modulation steps
      DO ISTP = 1, NSTPMD

C        Change range modulator thickness
         IF (IVOLMD.NE.0) THEN
            THICVL(IVOLMD) = THICMD(ISTP)
         END IF

C        beta squared and charge squared
         BESQ = EKINBM*(EKINBM+2.*AMASBM)/(AMASBM+EKINBM)**2
         CHSQ = CHRGBM*CHRGBM

C        Transport the beam through beamline devices
         PARM1(KCHGPB) = CHRGBM                  ! charge
         PARM1(KMASPB) = AMASBM                  ! mass
         PARM1(KXOPB) = XYZBM(1)                 ! beam origin
         PARM1(KYOPB) = XYZBM(2)
         PARM1(KZOPB) = XYZBM(3)
         PARM1(KDXPB) = UVWBM(1)                 ! beam direction
         PARM1(KDYPB) = UVWBM(2)
         PARM1(KDZPB) = UVWBM(3)
         PARM1(KFLUPB) = FLUEBM*RATIMD(ISTP)     ! number of particles
         PARM1(KEKIPB) = EKINBM                  ! mean kinetic energy
         PARM1(KDISPB) = 0.                      ! distance from beam origin
         PARM1(KDEPPB) = 0.                      ! water equiv depth
         PARM1(KX1PB) = 0.                       ! integrated thickness 1
         PARM1(KX2PB) = 0.                       ! integrated thickness 2
```

各コモン変数の定義をおこなう

各関数のルックアップテーブルを初期化する

レンジモジュレーションのステップでループ

レンジモジュレータのディグレータとしての厚さを変更

β の二乗と電荷の二乗を計算しておく

PARM1 は親ビームのビームパラメータを含む配列

```
      PARM1(KAAPB) = 0.                          ! angle variance
      PARM1(KATPB) = 0.                          ! covariance
      PARM1(KTTPB) = 0.                          ! spatial variance
      PARM1(KRRPB) = (SEKBM/RKSTOP(CHSQ,BESQ,ZOVAMT(IWATMT)
     &  ,EXTEMT(IWATMT)))**2                     ! initial straggling
      LSPR = .TRUE.                              ! beam spread on
      IF (IVOLPB.GT.0) THEN
         NVOL = IVOLPB
      ELSE
         NVOL = NVOLVL
      END IF
      DO IVOL = 1, NVOL
         CALL KSTRAK(LSPR,IVOL,PARM1)
      END DO

C     Generate and transport subbeams
      IF (IVOLPB.GT.0) THEN
         ZPOS = ZPOSVL(IVOLPB)
         PHI = PHIVL(IVOLPB)
         SPHI = SIN(PHI)
         CPHI = COS(PHI)
         AA = PARM1(KAAPB)
         AT = PARM1(KATPB)
         TT = PARM1(KTTPB)
         IF (IVOLWB.GT.0) THEN
            ZW = ZPOSVL(IVOLWB) ! wobbling height
            AW = ANGLWB         ! wobbling angle
            DW = ABS(ZW-ZPOS)   ! wobbling distance
         ELSE
            AW = 0.
            DW = 0.
         END IF
         DO IV = 1, NXYPB(2)
            V = XY1PB(2)+DXYPB(2)*(REAL(IV)-0.5)
            DO IU = 1, NXYPB(1)
```

通常の輸送ではビーム拡大を可能にする

サブビーム生成面があればそこまで、なければ最後まで輸送する

物体単位でビーム輸送をする

サブビーム生成を行う

親ビームのパラメータ

ワブラーのパラメータ

```
U = XY1PB(1)+DXYPB(1)*(REAL(IU)-0.5)
R = SQRT(U*U+V*V)
IF (TT.GT.0.) THEN ! wobbling/scattered beam
   CALL KSWOBB(AW,DW,AA,AT,TT,R,FLUX,ANGL,SIGM)
ELSE                ! broad parallel beam option
   FLUX = 1./DXYPB(1)/DXYPB(2)/REAL(NXYPB(1)*NXYPB(2))
   ANGL = 0.
   SIGM = SQRT(AA)
END IF
PARM2(KCHGPB)= PARM1(KCHGPB)
PARM2(KMASPB)= PARM1(KMASPB)
PARM2(KXOPB) = CPHI*U-SPHI*V
PARM2(KYOPB) = SPHI*U+CPHI*V
PARM2(KZOPB) = ZPOS
IF (R.GT.0.) THEN
   PARM2(KDXPB) = ANGL*PARM2(KXOPB)/R
   PARM2(KDYPB) = ANGL*PARM2(KYOPB)/R
ELSE
   PARM2(KDXPB) = 0.
   PARM2(KDYPB) = 0.
END IF
PARM2(KDZPB) = SIGN(SQRT(1.-ANGL*ANGL),UVWBM(3))
PARM2(KDISPB)= 0.
PARM2(KFLUPB)= PARM1(KFLUPB)*FLUX*DXYPB(1)*DXYPB(2)
PARM2(KEKIPB)= PARM1(KEKIPB)
PARM2(KDEPPB)= PARM1(KDEPPB)
PARM2(KX1PB) = PARM1(KX1PB)
PARM2(KX2PB) = PARM1(KX2PB)
PARM2(KAAPB) = SIGM**2
PARM2(KTTPB) = (DXYPB(1)**2+DXYPB(2)**2)/24. ! minimal
IF (TT.GT.0.) THEN
   PARM2(KATPB) = AT/TT*PARM2(KTTPB) ! preserve FD
ELSE
   PARM2(KATPB) = 0.
END IF
```

二次元メッシュの各グリッドの位置

その点でのフラックス、ビーム方向、広がりを計算

TT を 0 にしたとき(最初の物体がサブビーム生成面のとき)、平行ビームを生成する

PARM2 はサブビームのビームパラメータを含む配列

ANGL からビーム方向ベクトルを計算

フラックスにピクセル面積をかけて粒子数を得る

積分パラメータは親のものに加算していく

ピクセル内に一様分布するとしてビームの広がりを与える

ビームの焦点距離を親ビームと同じにする

```
                  PARM2(KRRPB) = PARM1(KRRPB)
                  IVOL = IVOLPB
                  DO WHILE (IVOL.LT.NVOLVL                          サブビーム生成面から以後の物体にビームを輸送する
     &            .AND.PARM2(KEKIPB)*PARM2(KFLUPB).GT.0.)
                     IVOL = IVOL+1
                     IF (IVOLCO.NE.0.AND.IVOL.LE.IVOLCO) THEN       コリメータまではビーム拡大をしない
                        LSPR = .FALSE. ! do not spread before collimator
                     ELSE
                        LSPR = .TRUE.
                     END IF
                     CALL KSTRAK(LSPR,IVOL,PARM2)                   ビームを輸送するサブルーチンを呼ぶ
                  END DO
               END DO
            END DO
         END IF
      END DO ! range modulation step ISTP

C     User analysis routine
      CALL UKLAST                                                   線量分布の解析や出力を行う

      END
```

2.5. サブルーチン

2.5.1. ksinit.f

```
************************************************************************
      SUBROUTINE KSINIT ! initialize look-up tables                     メインプログラムでUKINIT後に呼ばれルックアップテーブルの初期化をさせるルーチン
*                                                                     *
      IMPLICIT NONE
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
************************************************************************

      REAL RKERF,RKIERF,RKRESR,RKEKIN,RKLET ! functions
      REAL R ! dummy varaiable
```

`      R = RKERF(-1.)  ! error function` `      R = RKIERF(-1.) ! integrated error function` `      R = RKRESR(-1.) ! residual range as a funciton of energy` `      R = RKEKIN(-1.) ! kinetic energy as a function of range` `      R = RKLET(-1.) ! depth-dose table` `      RETURN` `      END`	各関数についてはそれぞれのコードを参照

2.5.2. ksmear.f

`**********************************************************************` `      SUBROUTINE KSMEAR(N,XLOW,XSIG,Y,YS) ! smears a plot by sigmas` `*                                                                    *` `      IMPLICIT NONE` `      INTEGER N    ! <- Number of bins` `      REAL XLOW(*) ! <- Lower edge of each bin` `                   !    XLOW(N+1) must be given to give upper limit` `      REAL XSIG(*) ! <- Sigma X at each bin` `      REAL Y(*)    ! <- Average height of each bin` `      REAL YS(*)   ! -> Smeared (averaged) height of each bin` `*                                                                    *` `*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *` `**********************************************************************` `      INCLUDE 'kcmcon.i'` `      REAL RKERF          ! error function` `      REAL RKIERF         ! integrated error function` `      INTEGER I,J,IDIR` `      REAL SI,XI1,XI2,XI,SJ,XJ1,XJ2,XJ,WJI,DXI,DXJ,S11,S12,S21,S22,DXIJ` `      REAL*8 WI,YI` `C     Loop on bin of interest I` `      DO I = 1, N` `         IF (XLOW(I+1).LT.XLOW(I)) THEN`	可変幅ヒストグラムを各ビンで与えられるシグマに応じてガウシアンでスメアするルーチン ビンの数 各ビンの最小値 ただし、最大ビンの最大値を決めるため XLOW(N+1)が与えられなければならない 各ビンでのシグマ 各ビンの高さ スメアした後の各ビンの高さ YS を計算したいビンに着目する

```
            XLOW(I+1) = XLOW(I)
          END IF
          XI1 = XLOW(I)                                                    ビンの最小値
          XI2 = XLOW(I+1)                                                  最大値
          XI = 0.5*(XI1+XI2)                                               平均値
          DXI = XI2-XI1                                                    ビン幅
          SI = M_SQRT2*XSIG(I)                                             ルート2倍のシグマ

C         Self contribution (I -> I)                                       自ビンからの寄与
          IF (SI.LE.0.) THEN            ! delta function                   シグマが０ならばデルタ関数
            WI = DXI
          ELSE IF (DXI.LT.0.25*SI) THEN ! gaussian at center approximation ビン幅が小さいときはビンの中心を平均値とするガウス分布で近似
            WI = RKERF(0.5*DXI/SI)*DXI
          ELSE                          ! flat distribution                一般にはビン幅内でフラットに分布する階段関数にガウス分布をコンボリュートする
            WI = SI*RKIERF(DXI/SI)
          END IF                                                           ここでWIはビンＩに高さ１のコンテンツがあったとしてこれがビンＩに残る分の面積
          YI = Y(I)*WI                                                     ここでYIはビンＩに高さY(I)のコンテンツがあったとしてこれがビンＩに残る分の面積

C         Contributions from neighboring bins (J -> I)                     他ビンからの寄与
          DO IDIR = -1, 1, 2                                               自ビンに近いものから計算していくための探す方向
            J = I+IDIR
            WJI = 1.
            DO WHILE (J.GE.1.AND.J.LE.N.AND.WJI.GT.0.)                     遠くにいって、寄与がなくなったらそこまでで止める
              XJ1 = XLOW(J)
              XJ2 = XLOW(J+1)
              XJ = 0.5*(XJ1+XJ2)
              DXJ = XJ2-XJ1
              SJ = M_SQRT2*XSIG(J)
              DXIJ = MIN(ABS(XI2-XJ1),ABS(XI1-XJ2))                        ビンＪとビンＩの最短距離
              IF (SJ.LE.0..OR.DXIJ.GT.2.0*SJ) THEN ! out of range          これが非常に遠いときはもう止める
                WJI = 0.
              ELSE IF (DXIJ.GT.SJ) THEN ! far bin                          ある程度遠いとき
                WJI = DXI*DXJ/M_SQRTPI/SJ*EXP(-((XI-XJ)/SJ)**2)            ビンＪの中心を平均とするガウス分布で近似しビンＩの中心での値で近似
              ELSE IF (DXJ.LT.SJ) THEN  ! narrow bin                       ビンＪの幅が小さいとき
                WJI = 0.5*DXJ*(RKERF((XI2-XJ)/SJ)-RKERF((XI1-XJ)/SJ))      ビンＪの中心を平均とするガウス分布で近似しこれををビンＩの範囲で積分
```

```
            ELSE                          ! near and wide bin
               S11 = RKIERF((XI1-XJ1)/SJ)
               S12 = RKIERF((XI2-XJ1)/SJ)
               S21 = RKIERF((XI1-XJ2)/SJ)
               S22 = RKIERF((XI2-XJ2)/SJ)
               WJI = SJ*0.5*(S12+S21-S11-S22)*DXJ/DXI
            END IF
            WI = WI+WJI
            YI = YI+WJI*Y(J)
            J = J+IDIR
         END DO
      END DO
      YS(I) =YI/WI
   END DO

   RETURN
   END
```

一般には
階段関数をガウス分布でスメアし、これを積分

WJI はビン J に高さ 1 のコンテンツがあったとしてこれがビン I に入る分の面積
YJI はビン J に高さ Y(I)のコンテンツがあったとしてこれがビン I に入る分の面積
WI, YI にそれぞれを加える

スメアした高さは YI と WI の比

2.5.3. ksptdo.f

```
*********************************************************************
      SUBROUTINE KSPTDO(XYZO, UVW, ND, DISM, SLET, TSIG, KVOX)
*                                                                   *
*     Distribute beam dose in patient (Pol-beam association)        *
*                                                                   *
      IMPLICIT NONE
      REAL    XYZO(3) ! beam origin in patient frame
      REAL    UVW(3)  ! beam direction in patient frame
      INTEGER ND      ! number of steps or table size
      REAL    DISM(*) ! distances from beam origin
      REAL    SLET(*) ! LET's
      REAL    TSIG(*) ! beam spreads projected to one of lateral axes
      INTEGER KVOX(*) ! Points of interest on the beam path
*                                                                   *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)     *
*********************************************************************
```

ビームのヒストリー(輸送中の複数点でのビームパラメータのテーブル)を基にこのビームが近傍の計算点に寄与する線量を線量分布に加える

ビーム原点の患者座標系での位置
ビーム進行方向の患者座標系での表現
ヒストリーテーブルのデータ数
原点からの幾何学的距離
LET
ビームの広がり
通過したボクセル

```
INCLUDE 'kcmcon.i'
INCLUDE 'kcpati.i'
INCLUDE 'kcpenc.i'
INCLUDE 'kcphys.i'

REAL    RKINTP          ! interpolation function
INTEGER IKPXYZ          ! voxel number to index function

INTEGER MBIN            ! max number of bins
PARAMETER (MBIN=256)
INTEGER NBIN            ! number of bins
REAL    WBIN            ! bin width
REAL    SLBIN(MBIN)     ! SLET table
REAL    TTBIN(MBIN)     ! TSIG**2 table
INTEGER IBIN            ! bin number
REAL    BIN             ! (fractional bin number)-0.5

INTEGER ID              ! step number
INTEGER NBASE(3)        ! base numbers for indices
INTEGER IXYZ(3)         ! central axis voxel indices
REAL    DXYZ(3)         ! vector to Pol from beam origin
INTEGER IAX1,IAX2,IAX3  ! axis numbers
INTEGER I1,I2,I3        ! voxel indices for IAX1, IAX2, IAX3
INTEGER IVOX            ! voxel number
INTEGER IP1,IP2         ! temporary variables for IVOX
INTEGER N2              ! number of voxels at I2

REAL    DISM1           ! longitudinal distance to Pol from beam
REAL    RSQR1           ! radius squared to Pol from beam
REAL    SLET1           ! LET at DISM1
REAL    TT1             ! beam spread at DISM1
REAL    TTMIN           ! practical minimum for TT1
REAL    SIGT2           ! SIGTPH**2
```

高速計算用一時的なルックアップテーブルの定義

```
      INTEGER I,I1PREV,I2DIR,I2INI,I2FIN,I3DIR,I3INI,I3FIN
      REAL    X,Y

C     Base numbers
      NBASE(1) = 1
      NBASE(2) = NXYZPT(1)
      NBASE(3) = NXYZPT(1)*NXYZPT(2)

C     Determine the base axis
      IAX1 = 0
      X = 0.
      DO I = 1, 3
         Y = (UVW(I)/DXYZPT(I))**2
         IF (Y.GT.X) THEN
            X = Y
            IAX1 = I
         END IF
      END DO
      IAX2 = MOD(IAX1,3)+1
      IAX3 = MOD(IAX2,3)+1

C     Minimum beam spread squared
      IF (DSIZPH.LT.0.) THEN ! limitation due to voxel size
         TTMIN = (DXYZPT(1)**2*(1.-UVW(1)**2)
     &           +DXYZPT(2)**2*(1.-UVW(2)**2)
     &           +DXYZPT(3)**2*(1.-UVW(3)**2))/24.
      ELSE                   ! fixed dosimeter size
         TTMIN = DSIZPH**2
      END IF

C     Maximum beam spread squared of interest in unit of TT1
      SIGT2 = SIGTPH**2

C     Prepare lookup table for fast interpolation
      NBIN = MIN(MBIN,5*ND) ! 5 times number of steps should be enough
```

ボクセル番号とボクセルインデックスの変換用

最もインデックスが変化する軸を基本軸とする

基本軸でない軸

ビームの広がりの最小値として仮想線量計のサイズあるいはボクセルサイズをとる

横方向はSIGTPHシグマ以上は範囲外とするためのカットオフ

高速補間のためのルックアップテーブルの準備

```
      WBIN = (DISM(ND)-DISM(1))/REAL(NBIN)
      ID = 1
      DO I = 1, NBIN
         X = DISM(1)+WBIN*(REAL(I)-0.5) ! distance at center of bin
         DO WHILE (DISM(ID+1).LT.X)
            ID = ID+1
         END DO
         SLBIN(I) = MAX(0.,RKINTP(1,2,DISM(ID),SLET(ID),X))
         TTBIN(I) = MAX(TTMIN,RKINTP(1,2,DISM(ID),TSIG(ID),X)**2)
      END DO

C     Track the beam along the path
      I1PREV = 0
      DO ID = 1, ND
         IF (KVOX(ID).NE.0) THEN

C           Decompose voxel number into indices
            DO I = 1, 3
               IXYZ(I) = IKPXYZ(KVOX(ID),I)
            END DO

C           Search voxels in a plane at IXYZ(IAX1)
            I1 = IXYZ(IAX1)
            IF (I1.NE.I1PREV) THEN
               I1PREV = I1
               IP1 = 1+(I1-1)*NBASE(IAX1)
               DXYZ(IAX1) = (REAL(I1)-0.5)*DXYZPT(IAX1)
     &         +XYZ1PT(IAX1)-XYZ0(IAX1)

               DO I2DIR = 1, -1, -2
                  IF (I2DIR.EQ.1) THEN
                     I2INI = IXYZ(IAX2)
                     I2FIN = NXYZPT(IAX2)
                  ELSE
                     I2INI = IXYZ(IAX2)-1
```

LET
ビームの広がりの二乗

通過したボクセルをみていく

基本軸のインデックスが変化したとき

基本軸から計算されるボクセル番号
計算点(ボクセル中心)の基本軸の座標

通過したボクセルを中心に基本軸に垂直な平面内の各計算点の計算を行う
内側から両方向に計算していき遠くなりすぎたら止める

```
                  I2FIN = 1
               END IF

               DO I2 = I2INI, I2FIN, I2DIR                              第 2 軸でループ
                  N2 = 0
                  IP2 = IP1+(I2-1)*NBASE(IAX2)
                  DXYZ(IAX2) = (REAL(I2)-0.5)*DXYZPT(IAX2)
     &            +XYZ1PT(IAX2)-XYZ0(IAX2)

                  DO I3DIR = 1, -1, -2                                  第 3 軸についても同様
                     IF (I3DIR.EQ.1) THEN
                        I3INI = IXYZ(IAX3)
                        I3FIN = NXYZPT(IAX3)
                     ELSE
                        I3INI = IXYZ(IAX3)-1
                        I3FIN = 1
                     END IF

                     DO I3 = I3INI, I3FIN, I3DIR                        第 3 軸でループ
                        IVOX = IP2+(I3-1)*NBASE(IAX3)                   ボクセル番号がきまる
                        DXYZ(IAX3) = (REAL(I3)-0.5)*DXYZPT(IAX3)        計算点座標もきまる
     &                  +XYZ1PT(IAX3)-XYZ0(IAX3)

C                       Distance along beam axis from origin
                        DISM1 = DXYZ(1)*UVW(1)+DXYZ(2)*UVW(2)           ビーム原点からビーム軸に沿った距離
     &                  +DXYZ(3)*UVW(3)
                        IF (DISM1.GT.DISM(ND)) GOTO 11                  大きすぎたら次の I2 へ

C                       Calculate bin number for the lookup tables
                        BIN = (DISM1-DISM(1))/WBIN+0.5
                        IBIN = MAX(1,MIN(NBIN-1,INT(BIN+0.5)))          この距離に対するルックアップテーブルのビン番号

C                       Get TSIG from the look-up table
                        TT1 = TTBIN(IBIN)                               ビームの広がりの二乗を補間により求める
     &                  +(TTBIN(IBIN+1)-TTBIN(IBIN))*(BIN-REAL(IBIN))
```

```
C                         Radius squared from beam axis
                          RSQR1 = DXYZ(1)**2+DXYZ(2)**2+DXYZ(3)**2
     &                    -DISM1**2
                          IF (RSQR1.GT.TT1*SIGT2) GOTO 11

C                         Get SLET from the look-up table
                          SLET1 = SLBIN(IBIN)
     &                    +(SLBIN(IBIN+1)-SLBIN(IBIN))*(BIN-REAL(IBIN))

C                         Add dose (=dE/dZ/rho*d2P/dXdY) from this beam
                          N2 = N2+1
                          DOSEPT(IVOX) = DOSEPT(IVOX)
     &                    +SLET1*0.5/M_PI/TT1*EXP(-0.5*RSQR1/TT1)

                        END DO ! I3
   11                   CONTINUE
                      END DO ! I3DIR
                      IF (N2.LE.0) GOTO 12
                    END DO ! I2
   12               CONTINUE
                  END DO ! I2DIR

              END IF ! change in base axis
           END IF ! in patient
        END DO ! end of step ID

        RETURN
        END
```

ビーム軸からの半径の二乗

大きすぎたら次の I2 へ

LET を補間により求める

I2 でカウントした計算点数
このビームからの寄与をこの計算点の線量に加算する

次の I3 へ
I3 をスキップした場合はここに直接来る

I2 での計算点がない場合残りの I2 をスキップする
次の I2 へ
I2 をスキップしたらここに直接来る

次の通過ボクセルへ

2.5.4. ksptmt.f

```
************************************************************************
      SUBROUTINE KSPTMT(IVOX, ZOVA, EXTE, DENS, RADL, ABSL)
*                                                                      *
*     Get medium parameters in patient                                 *
```

患者ボクセルに対する物質データの取得

```
*                                                                      *
      IMPLICIT NONE
      INTEGER IVOX    !  <- voxel number
      REAL ZOVA       !  -> Z/A
      REAL EXTE       !  -> I
      REAL DENS       !  -> rho
      REAL RADL       !  -> X0
      REAL ABSL       !  -> Lambda1
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcpati.i'
      INCLUDE 'kcmate.i'

      IF (IVOX.EQ.0) THEN ! in the air gap
         ZOVA = ZOVAMT(IAIRMT)
         EXTE = EXTEMT(IAIRMT)
         DENS = DENSMT(IAIRMT)
         RADL = RADLMT(IAIRMT)
         ABSL = ABSLMT(IAIRMT)
      ELSE                ! in the compressed water
         ZOVA = ZOVAMT(IWATMT)
         EXTE = EXTEMT(IWATMT)
         DENS = DENSPT(IVOX)
         RADL = RADLMT(IWATMT)
         ABSL = ABSLMT(IWATMT)
      END IF

      RETURN
      END
```

ボクセル番号が 0 のときは空気とする

そうでなければ水とする
ただし、密度は/KCPATI/の DENSPT で与えられる

2.5.5. ksptst.f

```
************************************************************************
      SUBROUTINE KSPTST(XYZP,UVWP,IVOX,RESP)
```

患者座標系の位置、方向からボクセル番号を計算し、次のボクセルまでの距離を計算する

```
*                                                                      *
*     Calculate distance to the next boundary.                         *
*                                                                      *
      IMPLICIT NONE
      REAL XYZP(3)  ! <-  particle location in PT coordinate
      REAL UVWP(3)  ! <-  particle direction in PT coordinate
      INTEGER IVOX  !  -> voxel number
      REAL RESP     !  -> distance to next boundary
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcpati.i'
      INCLUDE 'kcmate.i'

      REAL BIG
      PARAMETER (BIG=1.0E30)

      INTEGER IKPVOX
      REAL XYZC(3),DXYZ(3),XYZB(3),R
      INTEGER I,IXYZ(3)

C     Voxel indices
      DO I = 1, 3
         IXYZ(I) = INT((XYZP(I)-XYZ1PT(I))/DXYZPT(I)+1.)
      END DO

C     Out of the voxel region, the whole voxel region is the box
      IF (IXYZ(1).LT.1.OR.IXYZ(1).GT.NXYZPT(1).OR.
     &    IXYZ(2).LT.1.OR.IXYZ(2).GT.NXYZPT(2).OR.
     &    IXYZ(3).LT.1.OR.IXYZ(3).GT.NXYZPT(3)) THEN
         DO I = 1, 3
            DXYZ(I) = ABS(0.5*REAL(NXYZPT(I))*DXYZPT(I))
            XYZC(I) = XYZ1PT(I)+DXYZ(I)
         END DO
```

ボクセルインデックスは簡単に求まる

インデックスが領域外なら患者領域全体を箱とする

```
         IVOX = 0

C     In the voxel region, the voxel is the box
      ELSE
         DO I = 1, 3                                                    ボクセル内なら、このボクセルを箱とする
            DXYZ(I) = ABS(0.5*DXYZPT(I))
            XYZC(I) = XYZ1PT(I)+(REAL(IXYZ(I))-0.5)*DXYZPT(I)
         END DO
         IVOX = IKPVOX(IXYZ(1),IXYZ(2),IXYZ(3))
      END IF

C     Translate to the box coordinate
      DO I = 1, 3                                                       箱の中心を原点とする座標へ移動する
         XYZB(I) = XYZP(I)-XYZC(I)
         IF (IVOX.NE.0.AND.ABS(XYZB(I)).GT.DXYZ(I)) THEN                数値的な誤差でボクセル外へ出てしまう場合はボクセル内に修正する
            XYZB(I) = SIGN(DXYZ(I),XYZB(I))
         END IF
      END DO

C     find minimum distance to the box surface
      RESP = BIG
      DO I = 1, 3                                                       箱の表面の3面を調べる
         IF (UVWP(I).NE.0.) THEN                                        (他の3面は進行方向の反対となる)
            IF (IVOX.EQ.0) THEN
               R = (SIGN(DXYZ(I),-UVWP(I))-XYZB(I))/UVWP(I) ! from outside   ボクセル外なら外側から
            ELSE
               R = (SIGN(DXYZ(I),UVWP(I))-XYZB(I))/UVWP(I)  ! from inside    ボクセル内なら内側から
            END IF
            IF (R.GE.0..AND.R.LT.RESP) THEN
               RESP = R                                                 距離の最小となる面を通過する
            END IF
         END IF
      END DO

      RETURN
```

END	

2.5.6. kstrak.f

**********	ビーム１本について１つの物体およびその前の空気層を輸送する
SUBROUTINE KSTRAK(LSPR,IVOL,P) ! transport a beam	
* *	
IMPLICIT NONE	
LOGICAL LSPR ! <- Flag to enable beam spread	ボーラス・コリメータの順でビーム輸送するときの特別な取り扱い
INTEGER IVOL ! <- next volume number	
REAL P(*) ! <-> beam parameter array	ビームパラメータを含む配列
* *	
* 19970725 - N. Kanematsu (Mitsubishi Electric Corporation) *	
**********	
INCLUDE 'kcmcon.i'	
INCLUDE 'kcphys.i'	
INCLUDE 'kcvolu.i'	
INCLUDE 'kcwobb.i'	
INCLUDE 'kcmate.i'	
INCLUDE 'kcmodu.i'	
INCLUDE 'kccoll.i'	
INCLUDE 'kcbolu.i'	
INCLUDE 'kcpati.i'	
INCLUDE 'kcpenc.i'	
C External functions	
REAL*8 DKAA	
REAL RKESTR,RKSTOP,RKDFDR,RKENUC,RKLET,RKRESR,RKEKIN	
C Arrays to store beam parameter histories	
INTEGER MD ! max number of points	ビームパラメータのヒストリーを一時的にためておくための配列
PARAMETER (MD=512)	
INTEGER ND ! number of points	
INTEGER ID ! max point number with dose	
INTEGER KVOX(MD) ! voxels on the path	

```
      REAL DISM(MD),DEPM(MD),WDEP(MD) ! distance, depth, thickness
      REAL ULET(MD),SLET(MD)          ! unsmeared and smeared LET
      REAL RSIG(MD),TSIG(MD)          ! straggling and spread

C     Other temporary variables
      CHARACTER*4 CTYP
      INTEGER IREG,IMAT,IBOL,IVOX,I,J,IX,IY,IREG2
      REAL CHSQ,AMAS,BESQ,RESI,STEP,ZPOS,PHI,THIC,ZOVA,EXTE,CORR
     &,DENS,RADL,ABSL,XYZ(3),UVW(3),DELT,RESP,DELD,DELW,STOP,SCAT
     &,WER,CPHI,SPHI,ELOS,ENUC,FABS,DOSE,RESR,TV,EK,FL,EKO,FLO
     &,XL,YL,XV,YV,ATO,TTO,RRO,PQR(3),PQRO(3),PQRU(3),XLOW(MD)

C     Cumulative parameters
      REAL*8 AA,TT,AT,RR,PX1,PX2,DIST,DEPT,AAO,AACO

C     Safety margin for media boundary
      REAL EPS                                          次の物体まで確実に輸送するためのステップへの付加量
      PARAMETER (EPS=0.0001)

      FL = P(KFLUPB)
      EK = P(KEKIPB)
      IF (EK*FL.LE.0.) RETURN

      RESR = RKRESR(EK)       ! residual range in g/cm^2 H2O    ここからの残飛程
      CHSQ = P(KCHGPB)**2
      AMAS = P(KMASPB)                                  配列では使いにくいので普通の変数に入れる
      DIST = P(KDISPB)
      DEPT = P(KDEPPB)
      PX1  = P(KX1PB)
      PX2  = P(KX2PB)
      AA   = P(KAAPB)
      AT   = P(KATPB)
      TT   = P(KTTPB)
      RR   = P(KRRPB)
      AACO = AA-DKAA(PX1,PX2) ! correction if AA unequals to DKAA   角度の分散に散乱以外の効果があるときの補正項
```

```
      CTYP = CTYPVL(IVOL)      ! volume type
      ZPOS = ZPOSVL(IVOL)      ! height
      THIC = THICVL(IVOL)      ! thickness
      PHI  = PHIVL(IVOL)       ! rotation
      IMAT = IMATVL(IVOL)      ! material
      DO I = 1, 3
         UVW(I) = P(KDXPB+I-1)
         XYZ(I) = P(KXOPB+I-1)+P(KDISPB)*UVW(I)
      END DO
      XV = XYZ(1)+UVW(1)/UVW(3)*(ZPOS-XYZ(3)) ! location in VL coord
      YV = XYZ(2)+UVW(2)/UVW(3)*(ZPOS-XYZ(3))
      TV = SQRT(XV**2+YV**2)
      CPHI = COS(PHI)
      SPHI = SIN(PHI)
      XL = CPHI*XV+SPHI*YV ! location in local coord
      YL =-SPHI*XV+CPHI*YV
      IREG2 = 2

      IF (CTYP.EQ.CWOBVL) THEN      ! wobbler, do nothing
      ELSE IF (CTYP.EQ.CDEGVL) THEN ! degrader, do nothing
      ELSE IF (CTYP.EQ.CMODVL) THEN ! range modulater, do nothing
      ELSE IF (CTYP.EQ.CCOLVL) THEN ! collimator
         IMAT = IAIRMT
         IX = INT((XL-XY1CO(1))/DXYCO(1)+1.)
         IY = INT((YL-XY1CO(2))/DXYCO(2)+1.)
         IF (IX.LT.1.OR.IX.GT.NXYCO(1)
     &   .OR.IY.LT.1.OR.IY.GT.NXYCO(2)) THEN
            EK = 0.
            FL = 0.
         ELSE IF (.NOT.OPENCO(IX,IY)) THEN
            EK = 0.
            FL = 0.
         END IF
      ELSE IF (CTYP.EQ.CBOLVL) THEN ! bolus
```

ビーム座標系(KCVOLU)での位置、方向

物体中心の高さでの位置

ボーラスあるいはコリメータのローカル座標系での表現

物体の種類に応じた処理

コリメータの場合はビームが通過するかどうかを調べ、通過すれば空気とし、しない場合は直ちに止める

```
          IBOL = ISUBVL(IVOL)
          IX = INT((XL-XY1BO(1,IBOL))/DXYBO(1,IBOL)+1.)
          IY = INT((YL-XY1BO(2,IBOL))/DXYBO(2,IBOL)+1.)
          IF (IX.GE.1.AND.IX.LE.NXYBO(1,IBOL).AND.
     &        IY.GE.1.AND.IY.LE.NXYBO(2,IBOL)) THEN
             ZPOS = ZPOS+0.5*HIGHBO(IX,IY,IBOL)
             THIC = ABS(HIGHBO(IX,IY,IBOL))
          END IF
       ELSE IF (CTYP.EQ.CPATVL) THEN ! patient
          IMAT = IWATMT
          DO I = 1, 3
             PQRO(I) = 0.
             PQRU(I) = 0.
             PQR(I) = 0.
             DO J = 1, 3
                PQRO(I) = PQRO(I)+RV2PPT(I,J)*(P(KXOPB+J-1)-XYZOPT(J))
                PQRU(I) = PQRU(I)+RV2PPT(I,J)*UVW(J)
                PQR(I)  = PQR(I) +RV2PPT(I,J)*(XYZ(J)-XYZOPT(J))
             END DO
          END DO
       ELSE IF (CTYP.EQ.CPENVL) THEN ! subbeam generator
          IREG2 = 1 ! only air gap
       ELSE
          WRITE(6,*) 'KSTRAK> Invalid volume: IVOL,CTYP = '
     &    ,IVOL,CTYP
          IREG2 = 0 ! skip this volume
       END IF

C      Region loop (1 for air gap, 2 for the volume)
       IREG = 1
       DO WHILE (IREG.LE.IREG2.AND.EK*FL.GT.0.)
          IF (IREG.EQ.1) THEN ! in the air gap before the volume
             RESI = (ZPOS-XYZ(3)-SIGN(THIC/2.,UVW(3)))/UVW(3)
             ZOVA = ZOVAMT(IAIRMT)
             EXTE = EXTEMT(IAIRMT)
```

ボーラスの場合はフラット面での位置からピクセルを求め、所定の厚さを与える。

患者の場合は患者座標系での原点、位置、方向を求めておく

サブビーム生成面は実体なし

空気層を輸送し次に本体

RESI はこの物体または空気を出るまでの距離

```
            DENS = DENSMT(IAIRMT)
            RADL = RADLMT(IAIRMT)
            ABSL = ABSLMT(IAIRMT)
         ELSE                  ! in the volume
            RESI = (ZPOS-XYZ(3)+SIGN(THIC/2.,UVW(3)))/UVW(3)
            ZOVA = ZOVAMT(IMAT)
            EXTE = EXTEMT(IMAT)
            DENS = DENSMT(IMAT)
            RADL = RADLMT(IMAT)
            ABSL = ABSLMT(IMAT)
         END IF

C        Check volume order/overlaps
         IF (RESI.LT.-2.*EPS) THEN
            WRITE(6,*) 'KSTRAK> Invalid volume order '
     &      ,IVOL,IREG,' '//NAMEVL(IVOL)//' ',RESI
         END IF

C        Track though the medium
         RESP = 0.                ! distance to next PT voxel
         ND = 0                   ! number of data points in patient
         DO WHILE (RESI.GT.0.)
            AAO = AA
            ATO = AT
            TTO = TT
            RRO = RR
            EKO = EK
            FLO = FL
            STEP = RESI+EPS

C           For patient, step is limited within voxel
            IF (CTYP.EQ.CPATVL.AND.IREG.EQ.2) THEN
               IF (RESP.LE.0.) THEN
                  CALL KSPTST(PQR,PQRU,IVOX,RESP)
                  CALL KSPTMT(IVOX,ZOVA,EXTE,DENS,RADL,ABSL)
```

前ステップまでの値を記憶しておく

とりあえず物体全体を1ステップで動くことを考える

患者の場合下部構造としてボクセルがある

境界までの距離を得る
ボクセルの物質データを得る

END IF	
STEP = MIN(STEP, RESP+EPS)	ステップはボクセルの境界までとなる
END IF	
C Kinetic energy is allowed to be zero in patient	ストラグリングを含めるので患者内では飛程以降の点も計算する必要がある
IF (FLO*EKO.GT.0.) THEN	そうでない場合
C Calculate initial stopping power	
BESQ = EKO*(EKO+2.*AMAS)/(AMAS+EKO)**2 ! beta squared	ステップ開始時のβ二乗とストッピングパワー
STOP = RKSTOP(CHSQ, BESQ, ZOVA, EXTE)	
IF (ZOVA.NE.ZOVAMT(IWATMT)	
& .OR.EXTE.NE.EXTEMT(IWATMT)) THEN	
WER = STOP/RKSTOP(CHSQ, BESQ, ZOVAMT(IWATMT)	g/cm^2 当たりの水等価比
& ,EXTEMT(IWATMT)) ! water equivalent ratio	
ELSE	
WER = 1.	水ならば当然 1
END IF	
C Step size under limitations	物体の幾何学的サイズに加え、運動エネルギー変化率からの条件、最大ステップサイズ、最小ステップサイズの条件を課して、ステップサイズを決める
STEP = MIN(STEP, STMAPH	
& ,MAX(EPS, STMIPH, EKO/STOP/DENS*RELKPH))	
C Angle corrected path length for this step	
DELT = (1.+AAO)*STEP	ステップに角度分散の効果の補正をしてパスレングスを求める
DELD = DELT*DENS ! path length in g/cm^2	
DELW = DELD*WER ! path length in g/cm^2 H2O	
IF (DELW.GT.RESR-EPS) THEN	
STEP = STEP*RESR/DELW	
DELT = DELT*RESR/DELW	
DELD = DELD*RESR/DELW	
DELW = RESR	
END IF	
C Calculate energy loss and update EK	
EK = RKEKIN(MAX(0., RESR-DELW))	残飛程から平均運動エネルギーを求める

```
      CORR = MIN(0.5,0.5*STOP*DELD/EKO) ! correction term
      CORR = 0
      IF (DELW.LT.1.E-2) THEN ! too small for interpolation
         ELOS = (1.+CORR)*STOP*DELD
      ELSE
         ELOS = EKO-EK
      END IF
      IF (EK.LE.ECUTPH.OR.DELW.GE.RESR) THEN
         ELOS = EKO
         EK = 0.
      END IF

C     Calculate nuclear interactions and update FL
      IF (INUCPH.EQ.0.OR.ILETPH.NE.1) THEN
         FABS = 0.
         ENUC = 0.
      ELSE
C        Absorption probability
         FABS = MIN(1.,RKDFDR(RESR)*DELW*ABSLMT(IWATMT)/ABSL)
         ENUC = FABS*RKENUC(EKO)
      END IF
      FL = FLO*(1.-FABS)

C     Calculate unsmeared LET in MeV/g cm^2 in water
      DOSE = (ELOS+ENUC)*FLO/DELW*(DELT/STEP)

C     Calculate scattering and update PX1,PX2,AA,AT,TT,RR
      IF (ISCAPH.NE.0) THEN
         PX1 = PX1+DBLE(DELD/RADL*CHSQ
     &      *(1.+2.*CORR)/(BESQ*(AMAS+EKO))**2)
         IF (ISCAPH.EQ.1) THEN
            PX2 = PX2+DBLE(DELD/RADL)
         ELSE IF (ISCAPH.EQ.2) THEN
            PX2 = PX2+DBLE(DELD/RADL*CHSQ*(1.+CORR)/BESQ)
         END IF
```

ステップ内でのエネルギー変化の効果の補正項

あまり小さいステップではこの方が精度が良い

通常はエネルギーの差から線量への寄与を求める

カットオフエネルギー以下になった場合の処理

核反応の効果を含めないか既に含まれている場合

1つの陽子が吸収される確率
陽子が吸収された場合、落とすエネルギー

ストラグリングの効果を含まない水中でのLET

散乱角計算用のパラメータ1

パラメータ2(ハイランドの式)

パラメータ2(リンチ=ダールの式)

```
C                  Update parameters AA, AT, TT, RR
                   AA = DKAA(PX1,PX2)+AACO
                   SCAT = (AA-AAO)/DBLE(STEP)
                   IF (LSPR) THEN ! scattering propagation
                      AT = AT+DBLE(AAO*STEP)+DBLE(SCAT/2.*STEP**2)
                      TT = TT+DBLE(2.*ATO*STEP)+DBLE(AAO*STEP**2)
     &                +DBLE(SCAT/3.*STEP**3)
                   ELSE            ! fan beam effect only
                      TT = TT+DBLE(ATO*STEP)
                   END IF
                   IF (ISTRPH.GT.0.AND.TTO.GT.0.) THEN
                      RR = (SQRT(RR)+M_SQRT2*(AAO-ATO*ATO/TTO)*DELW)**2
                   END IF
                END IF

C                  Calculate straggling and update RR
                   IF (ISTRPH.EQ.1) THEN
                      RR = RR+DBLE((WER/(1.+CORR)/STOP)**2
     &                *RKESTR(CHSQ,BESQ,ZOVA)*DELD)
                   ELSE IF (ISTRPH.EQ.2) THEN
                      RR = (SQRT(RR)+DBLE(0.011*DELW))**2
                   END IF

             ELSE ! FLO*EKO

C                  Set step size to 0.5% of depth if EK < 0.
                   STEP = MIN(STEP,MAX(STMIPH,0.005*DEPT/DENS))
                   TT = TT+DBLE(ATO*STEP)
                   DOSE = 0.
                   DELT = (1.+AAO)*STEP
                   DELD = DELT*DENS ! path length in g/cm^2
                   DELW = DELD*WER  ! path length in g/cm^2 H2O
             END IF

C            Calculate dose in patient
```

散乱角は近似式で得る

フェルミ=アイジス理論による散乱効果の計算

散乱効果によるストラグリングの増加(実は無視できるほど小さい)

ガウシアン理論によるエネルギー損失のばらつきによるストラグリングの増加

水等価深さの 1.1%とする場合

飛程以後のビーム輸送

飛程の 0.5%ステップとする(ストラグリングの約半分)

```
      IF (CTYP.EQ.CPATVL.AND.IVOX.NE.0) THEN
         IF (ND.GE.MD) THEN
            WRITE(6,*) 'KSTRAK> too many data points.'
         ELSE
            ND = ND+1
            KVOX(ND) = IVOX                ! voxel number
            WDEP(ND) = DELW                ! bin width
            DISM(ND) = DIST+DBLE(0.5*STEP) ! mid point in cm
            DEPM(ND) = DEPT+DBLE(0.5*DELW) ! mid point in cmH2O
            ULET(ND) = DOSE                ! unsmeared LET
            IF (ILETPH.NE.1) THEN          ! smeared LET
               SLET(ND) = FLO*RKLET(DEPM(ND))*(DELT/STEP)
            END IF
            TSIG(ND) = SQRT(TTO) ! lateral spread in cm
            RSIG(ND) = SQRT(RRO) ! straggling in g/cm^2 H2O
            IF (DOSE.GT.0.) ID = ND ! last energy deposit point
         END IF
      END IF ! in patient

C     Proceed one step
      DIST = DIST+DBLE(STEP)
      RESI = RESI-STEP
      RESP = RESP-STEP
      DEPT = DEPT+DBLE(DELW)
      RESR = RESR-DELW
      DO I = 1, 3
         XYZ(I) = XYZ(I)+UVW(I)*STEP
      END DO
      IF (CTYP.EQ.'PATI') THEN
         DO I = 1, 3
            PQR(I) = PQR(I)+PQRU(I)*STEP
         END DO
      END IF

C     Kill this beam when unnecessary
```

患者領域内では線量を計算するのでヒストリーを残す

LET をルックアップテーブルで計算する場合

ブラッグピーク位置を覚えておく

ビームをステップ分輸送する
原点からの距離
物体中の残距離
ボクセル中の残距離
積分水等価深さ
残飛程

ビーム座標系での位置

患者座標系での位置

IF (EK*FL.LE.0.) THEN IF (IREG.EQ.2.AND.CTYP.EQ.CPATVL.AND.ND.GT.0) THEN IF (ILETPH.NE.1) THEN IF (SLET(ND).LT.SLET(1)*0.01) THEN RESI = 0. END IF ELSE IF (REAL(DEPT).GT.DEPM(ID)+SIGRPH*RSIG(ID)) THEN RESI = MIN(RESI,RSIG(ID)/DENS) END IF ELSE RESI = 0. IREG = IREG2 END IF END IF	基本的には運動エネルギーがなくなるか、粒子数がなくなれば直ちにとめるが患者内ではストラグリングの効果等でその先まで計算する必要がある ストラグリングを入れない場合は計算されるLETが十分小さくなったら止める ストラグリングを入れる場合はSIGRPHで指定される範囲プラス1シグマを超えたら止める
END DO ! WHILE (RESI.GT.0.)	物体を抜けるまで繰り返す
IREG = IREG+1 END DO ! WHILE (IREG.LE.IREG2.AND.EK*FL.GT.0.)	空気層と物体を計算する
C In patient, calculate dose IF (CTYP.EQ.CPATVL.AND.ND.GT.0) THEN	患者の場合線量分配をする
C Smear depth-dose curve (LET given in MeV/g cm^2 H2O) IF (ILETPH.EQ.1.AND.ISTRPH.GT.0) THEN DO I = 1, ND XLOW(I) = DEPM(I)-0.5*WDEP(I) ! low edge END DO CALL KSMEAR(ND-1,XLOW,RSIG,ULET,SLET) END IF	まだ入っていない場合、ストラグリングの効果をLETに入れる ULETをぼかしてSLETを計算する
C Cut off unnecessary data DO WHILE (SLET(ND).LE.0.) ND = ND-1 END DO	ゼロデータ点は必要なし

```
C         Distribute dose in patient
          CALL KSPTDO(PQRO, PQRU, ND, DISM, SLET, TSIG, KVOX)

        END IF ! CTYP

C       Return updated parameters
        P(KFLUPB) = FL
        P(KEKIPB) = EK
        P(KDISPB) = DIST
        P(KDEPPB) = DEPT
        P(KX1PB)  = PX1
        P(KX2PB)  = PX2
        P(KAAPB)  = AA
        P(KATPB)  = AT
        P(KTTPB)  = TT
        P(KRRPB)  = RR

        RETURN
        END
```

ビーム近傍の計算点に線量を付加する

アップデートしたビームパラメータを返す

2.5.7. kswobb.f

```
************************************************************************
      SUBROUTINE KSWOBB(AW, DW, AA, AT, TT, R, FLUX, ANGL, SIGM)
*                                                                      *
*     Calculate spreaded beam distribution by Wobbler method           *
*                                                                      *
      IMPLICIT NONE
      REAL AW       ! <-  wobbling angle [rad]
      REAL DW       ! <-  wobbling distance [cm]
      REAL AA       ! <-  mean projected angle squared [rad^2]
      REAL AT       ! <-  mean projected angle*displacement [cm rad]
      REAL TT       ! <-  mean projected displacement squared [cm^2]
      REAL R        ! <-  distance to beam axis [cm]
      REAL FLUX     !  -> Probability per unit area [cm^-2]
```

ワブラー法による拡大ビームの粒子分布を計算するルーチン

目的の位置のビーム軸からの距離

親粒子１つあたりで、この位置に単位面積あたりに入る確率

```
      REAL ANGL    !  -> average subbeam angle (in r direcion) [rad]
      REAL SIGM    !  -> sigma of subbeam angle [rad]
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcmcon.i'

      REAL*8 DKBESI    ! modified bessel function
      REAL*8 BESI0     ! modified Bessel funciton of order 0 at A
      REAL*8 DR        ! argument for DKBESI
      REAL*8 SS        ! argument for EXP
      REAL RW          ! wobbling radius
      REAL B1          ! ratio of order 1 divided by order 0
      REAL B2          ! ratio of order 2 divided by order 0

      RW = AW*DW

      DR = R*RW/TT
      BESI0 = DKBESI(0,DR)
      B1 = DKBESI(1,DR)/BESI0
      B2 = DKBESI(2,DR)/BESI0

      SS = -0.5*(R*R+RW*RW)/TT
      FLUX = 0.5D0/(M_PI*TT)*(EXP(SS)*BESI0)
      ANGL = R*AT/TT-(RW*AT/TT-AW)*B1
      SIGM = SQRT(AA-AT*AT/TT+(RW*AT/TT-AW)**2*((1.+B2)/2.-B1*B1))

      RETURN
      END
```

親ビーム軸から測った子ビームの角度(軸対称のため半径方向となる)
子ビーム角度のガウシアン標準偏差

これらの解析的計算式は外村氏のコード及びレポートに基づいている

2.6. INTEGER 関数

2.6.1. ikpvox.f

```
************************************************************************
      INTEGER FUNCTION IKPVOX(IX,IY,IZ)
```

ボクセルインデックスからボクセル番号を求める整数関数

```
*                                                                     *
*     Calculates voxel number from three voxel indices                *
*                                                                     *
      IMPLICIT NONE
      INTEGER IX ! X voxel index
      INTEGER IY ! Y voxel index
      INTEGER IZ ! Z voxel index
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
***********************************************************************

      INCLUDE 'kcpati.i'

      IKPVOX = IX+NXYZPT(1)*(IY-1+NXYZPT(2)*(IZ-1))

      RETURN
      END
```

わざわざボクセル番号なんてものを定義しないで 3 つのインデックスだけでボクセルを表しても良かったが、いまは歴史的な理由でこうなっているので簡単に変換できる関数を用意した

2.6.2. ikpxyz.f

```
***********************************************************************
      INTEGER FUNCTION IKPXYZ(IVOX,IXYZ)
*                                                                     *
*     Extract a voxel index from voxel number                         *
*                                                                     *
      IMPLICIT NONE
      INTEGER IVOX ! voxel number
      INTEGER IXYZ ! 1 to 3 to specify X to Z
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
***********************************************************************

      INCLUDE 'kcpati.i'

      INTEGER N,I
```

ボクセル番号からインデックスを取り出す整数関数

```
      N = 1
      DO I = 1, IXYZ-1
         N = N*NXYZPT(I)
      END DO
      IKPXYZ = MOD((IVOX-1)/N,NXYZPT(IXYZ))+1

      RETURN
      END
```

2.7. REAL 関数

2.7.1. rkdfdr.f

```
************************************************************************
      REAL FUNCTION RKDFDR(RESR)
*                                                                      *
*     Returns 1/F dF/dR for water, where                               *
*     F: fluence                                                       *
*     R: residual range measured in g/cm^2                             *
*                                                                      *
*     Ref: Lee and Webb, Phys. Med. Biol. 38(1993)989-998              *
*     where F is parametrized by F(R) = F(0)(1+0.012R)                 *
*                                                                      *
      IMPLICIT NONE
      REAL RESR ! residual range in g/cm^2 of water
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      RKDFDR = 0.012/(1.+0.012*RESR)

      RETURN
      END
```

水中での核反応により陽子 1 個、1cm あたり吸収される確率を与える

この論文によると、実験的に、陽子数は深さとともに直線的に減少する。
その減少率は飛程での陽子数を 1 とすると 0.012/cm である。

2.7.2. rklet.f

```
************************************************************************
      REAL FUNCTION RKLET(DEPT)
```

LET に対して実験データあるいは計算したルックアップテーブルを使う場合に参照する LET を与える実数関数

```
*                                                                      *
*     LET in water [MeV/g cm^2] as a function of depth                 *
*                                                                      *
      IMPLICIT NONE
      REAL DEPT ! depth in g/cm^2 H2O measured from beam source
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcbeam.i'
      INCLUDE 'kcmate.i'
      INCLUDE 'kcphys.i'
      INCLUDE 'kcdedo.i'

      REAL RKSTOP,RKDFDR,RKENUC,RKESTR,RKINTP,RKRESR,RKEKIN,RKINT2
      INTEGER N
      PARAMETER (N=512)
      REAL XMIN,Y(N),STEP
      DATA STEP/0./
      SAVE Y,XMIN,STEP

      REAL E(N),XLOW(N),YO(N),EKIN,RESR,FLUE,CORR,ELOS,BESQ,CHSQ
     &,XMID,RSTR,F,ENUC
      INTEGER I

C     Initialization
      IF (STEP.EQ.0..OR.DEPT.LT.0.) THEN
         IF (ILETPH.EQ.3) THEN ! depth-dose given by /KCDEDO/
            CORR = RKRESR(EKINBM)-RKRESR(EKINDD)
            XMIN = DEPTDD(1)+CORR
            STEP = (DEPTDD(NDDD)-DEPTDD(1))/REAL(N)
            DO I = 1, N
               XMID = XMIN+STEP*(REAL(I)-0.5)
               Y(I) = RKINTP(2,NDDD,DEPTDD,DOSEDD,XMID-CORR)
            END DO
```

ルックアップテーブルの定義

実験データを使う場合
ビームエネルギーが実験と異なる場合の補正

高速計算のため実験データを補間して等分割点での LET を求める

```
      ELSE                    ! theoretical calculation
         CHSQ = CHRGBM**2
         EKIN = EKINBM
         RESR = RKRESR(EKIN)
         XMIN = 0.
         STEP = RESR*1.2/REAL(N)
         FLUE = 1.
         RSTR = 0.
         DO I = 1, N
            XLOW(I) = XMIN+STEP*REAL(I-1)
            IF (EKIN.GT.0.) THEN
               BESQ = EKIN*(EKIN+2.*AMASBM)/(AMASBM+EKIN)**2
               RESR = MAX(0.,RESR-STEP)
               ELOS = EKIN-RKEKIN(RESR)
               IF (EKIN-ELOS.LT.ECUTPH.OR.RESR.EQ.0.) THEN
                  ELOS = EKIN
               END IF
               IF (ISTRPH.EQ.1) THEN
                  RSTR = RSTR
     &               +RKESTR(CHSQ,BESQ,ZOVAMT(IWATMT))*STEP
     &               /RKSTOP(CHSQ,BESQ,ZOVAMT(IWATMT),EXTEMT(IWATMT))**2
                  E(I) = SQRT(RSTR)
               ELSE IF (ISTRPH.EQ.2) THEN
                  E(I) = 0.011*(XLOW(I)+0.5*STEP)
               END IF
               IF (INUCPH.EQ.1) THEN
                  F = MIN(1.,RKDFDR(RESR)*STEP) ! absorption prob
                  ENUC = RKENUC(EKIN)*F         ! absorbed energy
               ELSE
                  F = 0.
                  ENUC = 0.
               END IF
               YO(I) = (ELOS+ENUC)/STEP*FLUE ! average LET
               EKIN = EKIN-ELOS
               FLUE = FLUE*(1.-F)
```

理論計算の場合

飛程の 1.2 倍まで計算する

同様に等分割点での LET をテーブルにする

エネルギー損失は残飛程－エネルギー関係から求める

エネルギーストラグリングの計算

核反応の効果

LET の計算

```
               ELSE
                  Y0(I) = 0.
                  E(I) = SQRT(RSTR)
               END IF
            END DO
            CALL KSMEAR(N-1,XLOW,E,Y0,Y) ! straggling convolution
         END IF
      END IF

C     Lookup table interpolation
      RKLET = MAX(0.,RKINT2(N,XMIN,STEP,Y,DEPT))

      RETURN
      END
```

運動エネルギーがゼロでもしばらく計算する

ストラグリングでぼかして LET を得る

高速二次補間

2.7.3. rkdxde.f

```
************************************************************************
      REAL FUNCTION RKDXDE(EKIN)
*                                                                      *
*     returns -dX/dE in water [gH2O/cm^2/MeV]                          *
*     for Simpson's integration method to calculate range              *
*                                                                      *
      IMPLICIT NONE
      REAL EKIN ! kinetic energy [MeV]
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcbeam.i'
      INCLUDE 'kcmate.i'

      REAL RKSTOP,BESQ,CHSQ

      CHSQ = CHRGBM*CHRGBM
```

残飛程－エネルギー関係を求めるために行うシンプソン法数値積分のための非積分関数
水中のストッピングパワーの逆数を返す

```
      BESQ = EKIN*(EKIN+2.*AMASBM)/(EKIN+AMASBM)**2
      RKDXDE = 1./RKSTOP(CHSQ,BESQ,ZOVAMT(IWATMT),EXTEMT(IWATMT))

      RETURN
      END
```

2.7.4. rkekin.f

```
************************************************************************
      REAL FUNCTION RKEKIN(RESR)
*                                                                      *
*     Returns mean kinetic energy at given residual range              *
*                                                                      *
      IMPLICIT NONE
      REAL RESR ! residual range in water [gH2O/cm^2]
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcbeam.i'
      INCLUDE 'kcphys.i'

      REAL RKRESR,RKINT1

      INTEGER N,I
      PARAMETER (N=512)
      REAL XMIN,STEP,Y(N),EK,RS,EPS,RS1,EK1,EK2
      DATA STEP/0./
      DATA EPS/1.E-5/
      SAVE Y,XMIN,STEP

C     Initialization
      IF (STEP.EQ.0..OR.RESR.LT.0.) THEN
         XMIN = 0.
         STEP = (SQRT(RKRESR(EKINBM))-XMIN)/REAL(N)
         EK1 = ECUTPH
```

残飛程から運動エネルギーを求める関数

ルックアップテーブルの定義

ルックアップテーブルの初期化
残飛程のルートを等分割して対応する運動エネルギーをテーブルにする。

```
         DO I = 1, N
            RS1 = XMIN+STEP*(REAL(I)-0.5)
            EK2 = EKINBM
            DO WHILE (ABS(EK1-EK2)/(EK1+EK2).GT.EPS)
               EK = (EK1+EK2)/2.
               RS = SQRT(RKRESR(EK))
               IF (RS.GT.RS1) THEN
                  EK2 = EK
               ELSE
                  EK1 = EK
               END IF
            END DO
            Y(I) = EK ! kinetic energy at every SQRT(RESR) bin
         END DO
      END IF

      IF (RESR.LE.0.) THEN
         RKEKIN = 0.
      ELSE
         RKEKIN = MAX(0.,RKINT1(N,XMIN,STEP,Y,SQRT(RESR)))
      END IF

      RETURN
      END
```

残飛程 RS1 に対応する運動エネルギーは運動エネルギーから残飛程を求める関数 RKRESR から二分法で求める。

残飛程は常に正でゼロなら運動エネルギーもゼロ

残飛程のルートで高速補間

2.7.5. rkenuc.f

```
************************************************************************
      REAL FUNCTION RKENUC(EK)
*                                                                      *
*     Returns mean local energy deposition from nuclear interaction    *
*     The simplest formulation to reproduce plots in reference         *
*     Carlsson et al., Phys. Med. Biol. 42(1997)1033-1053.             *
*                                                                      *
      IMPLICIT NONE
```

1つの陽子の核反応から生ずる荷電二次粒子が落とす全エネルギーを与える

```
      REAL EK ! proton kinetic energy at interaction
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
***********************************************************************

      RKENUC = MAX(0.,EK-45.)

      RETURN
      END
```

カールソンの論文の彼らのモデル計算による二次粒子からの線量のプロットを再現する最も簡単な近似式。

2.7.6. rkerf.f

```
***********************************************************************
      REAL FUNCTION RKERF(X)
*                                                                     *
*     Fast look-up-table interpolation for error function ERF(X)      *
*                                                                     *
*     erf(x) = 2/sqrt(pi)*Integral[0->x](exp(-t*t)dt)                 *
*     Integral[x,inf](normal(t)dt) = (1-erf(x/sqrt(2)))/2             *
*     where normal(x) is the normal gaussian distribution function.   *
*                                                                     *
      IMPLICIT NONE
      REAL X
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
***********************************************************************

      REAL RKINT1,ERF
      INTEGER N,I
      PARAMETER (N=256)
      REAL Y(N),STEP,XMIN,XMAX,X1
      PARAMETER (XMAX=4.)
      DATA STEP/0./
      SAVE XMIN,Y,STEP

C     Initialization
```

誤差関数 ERF に相当する補間関数

関数の機能としては C、FORTRAN の標準拡張関数 erf と同等。プラットフォームによっては単純に erf を使用した方が早い場合が考えられる。その場合は RKERF の代わりに ERF を使用する。

```
      IF (STEP.EQ.0.) THEN
         XMIN = 0.
         STEP = (XMAX-XMIN)/REAL(N)
         DO I = 1, N
            X1 = XMIN+STEP*(REAL(I)-0.5)
            Y(I) = ERF(X1)
         END DO
      END IF

      X1 = ABS(X)
      IF (X1.LT.0.5*STEP) THEN
         RKERF = 2.*X*Y(1)/STEP
      ELSE
         RKERF = RKINT1(N,XMIN,STEP,Y,X1)
         IF (X.LT.0.) RKERF = -RKERF
      END IF

      RETURN
      END
```

Xを等分割し各点で誤差関数を計算しておく

誤差関数は反対称

0で0にするための修正

一次関数で補間

2.7.7. rkestr.f

```
************************************************************************
      REAL FUNCTION RKESTR(CHSQ,BESQ,ZOVA)
*                                                                      *
*     The lowest order calculation for ionization energy loss          *
*     fluctuation squared per unit path length  [MeV^2 g^-1 cm^2]      *
*                                                                      *
      IMPLICIT NONE
      REAL CHSQ ! particle charge squared
      REAL BESQ ! particle beta squared
      REAL ZOVA ! medium effective Z/A
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************
```

エネルギー損失の統計的なふらつきから単位物質長当たりエネルギーストラグリングの二乗を与える関数

```
      INCLUDE 'kcmcon.i'
      INCLUDE 'kcpcon.i'

      REAL FM
      PARAMETER (FM=4.*M_PI*AVOGAD*ERADI*ERADI*EMASS*EMASS)

      RKESTR = FM*CHSQ*ZOVA*(1.-0.5*BESQ)/(1.-BESQ)

      RETURN
      END
```

ある程度厚い物質ではエネルギーロスはガウス分布で近似され、これはその最低次の近似による分散の増加率を与える式

2.7.8. rkierf.f

```
************************************************************************
      REAL FUNCTION RKIERF(X)
*                                                                      *
*     Look-up-table interpolation for integrated error function        *
*                                                                      *
*     Ierf(x) = Integrate[0->x](erf(t)dt)                              *
*                                                                      *
      IMPLICIT NONE
      REAL X
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      REAL RKSIMP,RKERF,RKINT1
      EXTERNAL RKERF

      INTEGER N,I
      PARAMETER (N=256)
      REAL Y(N),STEP,XMIN,XMAX,X1,X0
      REAL*8 Y1
      PARAMETER (XMAX=4.)
      DATA STEP/0./
      SAVE XMIN,Y,STEP
```

誤差関数 erf の 0 からの積分値を与える関数

```
C     Initialization
      IF (STEP.EQ.0.) THEN
         XMIN = 0.
         STEP = (XMAX-XMIN)/REAL(N)
         X1 = 0.
         Y1 = 0.
         DO I = 1, N
            X0 = X1
            X1 = XMIN+STEP*(REAL(I)-0.5)
            Y1 = Y1+DBLE(RKSIMP(X0,X1,RKERF))
            Y(I) = Y1
         END DO
      END IF

      X1 = ABS(X)
      IF (X1.LT.0.5*STEP) THEN
         RKIERF = 2.*X1*Y(1)/STEP
      ELSE
         RKIERF = RKINT1(N,XMIN,STEP,Y,X1)
      END IF

      RETURN
      END
```

X を等分割し、各分割点間は誤差関数をシンプソン積分する。

この関数は対称

ゼロでゼロにするための修正

一次関数で補間

2.7.9. rkint1.f

```
************************************************************************
      REAL FUNCTION RKINT1(N,XMIN,STEP,Y,X)
*                                                                      *
*     Linear interpolation (returns interpolated y at x1)              *
*     for equal-sized bin histogram                                    *
*                                                                      *
      IMPLICIT NONE
      INTEGER N    ! number of data points
      REAL XMIN    ! lower edge of first bin
```

高速一次関数補間を行うための関数

補間に使う等ビン幅のヒストグラム的なデータ

```
      REAL STEP     ! bin size
      REAL Y(*)     ! y at mid point of each bin
      REAL X        ! x point of interest
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      REAL X1
      INTEGER I1

      X1 = (X-XMIN)/STEP+0.5            ! (fractional bin number)-0.5
      I1 = MIN(MAX(INT(X1+0.5),1),N-1) ! bin number
      RKINT1 = Y(I1)+(Y(I1+1)-Y(I1))*(X1-REAL(I1))

      RETURN
      END
```

高速線形補間計算

2.7.10. rkint2.f

```
************************************************************************
      REAL FUNCTION RKINT2(N,XMIN,STEP,Y,X)
*                                                                      *
*     Parabola  interpolation (returns interpolated y at x)            *
*     for equal-sized bin histogram                                    *
*                                                                      *
      IMPLICIT NONE
      INTEGER N     ! number of data points
      REAL XMIN     ! lower edge of first bin
      REAL STEP     ! bin size
      REAL Y(*)     ! y at mid point of each bin
      REAL X        ! x point of interest
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      REAL XI,DX
```

高速二次関数補間を行うための関数

補間に使う等ビン幅のヒストグラム的なデータ

```
      INTEGER I1,I2,I3

      XI = (X-XMIN)/STEP+0.5 ! (fractional bin number)-0.5
      I2 = INT(XI+0.5)       ! nearest bin number
      IF (I2.LE.1) THEN                                          ヒストグラムの端では線形補間
         RKINT2 = Y(1)+(Y(2)-Y(1))*(XI-1.)
      ELSE IF (I2.GE.N) THEN
         RKINT2 = Y(N)+(Y(N)-Y(N-1))*(XI-REAL(N))
      ELSE                                                       高速パラボラ補間計算
         I1 = I2-1
         I3 = I2+1
         DX = XI-REAL(I2)
         RKINT2 = Y(I2)+0.5*(Y(I3)-Y(I1)+(Y(I3)+Y(I1)-2.*Y(I2))*DX)*DX
      END IF

      RETURN
      END
```


2.7.11. rkintp.f

```
************************************************************************      可変間隔のデータ点セットを用いた線形あるいはパラボラ補間計算のための関数
      REAL FUNCTION RKINTP(IORD,N,X,Y,XO)
*                                                                     *
*     Linear/parabolic  interpolation (returns interpolated y at x0)  *
*                                                                     *
      IMPLICIT NONE
      INTEGER IORD ! order of interporation (1:linear or 2:parabollic)
      INTEGER N    ! number of data points
                   ! > 0 X's are in ascending order                    N 個の二次元データ点セットからある点での値を補間により求める
                   ! < 0 X's are in descending order                   データ点は X についてソートされている必要がある。
      REAL X(*)    ! x data points
      REAL Y(*)    ! y data points
      REAL XO      ! x point of interest
*                                                                     *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)       *
************************************************************************
```

```
      INTEGER I1,I2,I
      REAL X00,X01,X02,Y01,Y02,A,B

      I1 = 1
      I2 = ABS(N)
      DO WHILE (I2-I1.GT.1)
         I = (I1+I2)/2
         IF (X0*REAL(N).LT.X(I)*REAL(N)) THEN
            I2 = I
         ELSE
            I1 = I
         END IF
      END DO

      IF (IORD.LE.1) THEN                                          線形補間の場合
         RKINTP = Y(I1)+(Y(I2)-Y(I1))/(X(I2)-X(I1))*(X0-X(I1))
      ELSE
         I = MIN(N-1,MAX(2,I1+INT((X0-X(I1))/(X(I2)-X(I1))+0.5)))  パラボラ補間の場合
         X00 = X0-X(I)
         X01 = X(I-1)-X(I)
         X02 = X(I+1)-X(I)
         Y01 = Y(I-1)-Y(I)
         Y02 = Y(I+1)-Y(I)
         A = (Y01/X01-Y02/X02)/(X01-X02)
         B = Y01/X01-A*X01
         RKINTP = Y(I)+(A*X00+B)*X00
      END IF

      RETURN
      END
```

2.7.12. rkresr.f

```
************************************************************************      与えられた運動エネルギーから残飛程を求める補間関数
      REAL FUNCTION RKRESR(EKIN)
```

```
*                                                                      *
*     Returns mean residual range [g/cm^2 H2O] at given kinetic energy *
*     for particle given in /KCBEAM/                                   *
*                                                                      *
      IMPLICIT NONE
      REAL EKIN ! kinetic energy [MeV]
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcbeam.i'
      INCLUDE 'kcphys.i'

      REAL RKDXDE,RKSIMP,RKINT1
      EXTERNAL RKDXDE

      INTEGER N,I                                                    補間のためのルックアップテーブル
      PARAMETER (N=512)
      REAL XMIN,STEP,Y(N),EK1,EK2
      DATA STEP/0./
      SAVE XMIN,STEP,Y
      REAL*8 RS

      IF (STEP.EQ.0..OR.EKIN.LT.0.) THEN                             ルックアップテーブルの初期化
         XMIN = ECUTPH
         STEP = (EKINBM-XMIN)/REAL(N)                                ビームエネルギーからカットオフエネルギーまで等分割
         RS = 0.
         EK2 = XMIN
         DO I = 1, N
            EK1 = EK2
            EK2 = XMIN+STEP*(REAL(I)-0.5)
            RS = RS+DBLE(RKSIMP(EK1,EK2,RKDXDE))                     ストッピングパワーの逆数を積分して残飛程0からエネルギー損失を逆算する
            Y(I) = RS
         END DO
      END IF
```

```
      IF (EKIN.LE.ECUTPH) THEN
         RKRESR = 0.
      ELSE
         RKRESR = MAX(0., RKINT1(N, XMIN, STEP, Y, EKIN))
      END IF

      RETURN
      END
```

カットオフエネルギー以下では残飛程 0 とする

通常は高速線形補間

2.7.13. rksimp.f

```
************************************************************************
      REAL FUNCTION RKSIMP(A, B, FUNC)
*                                                                      *
*     Simpson Integration of function FUNC between [A, B]              *
*     I have no idea how this algorithm works.                         *
*     I just took it from Dr. Tomura's code.                           *
*                                                                      *
      IMPLICIT NONE
      REAL A    ! lower limit of the integration range
      REAL B    ! upper limit of the integration range
      REAL FUNC ! real function FUNC(X) to be integrated
      EXTERNAL FUNC
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INTEGER MAXI        ! maximum 2**MAXI divisions
      PARAMETER (MAXI=10)
      REAL E              ! max relative error
      PARAMETER (E=1.E-4)

      INTEGER INDEX
      REAL FAFB, XH, XHA, XI, XJ, XNEW, XIP, FNEWX, S, AA, BB
```

シンプソン法による数値積分

シンプソン法とは任意の関数を二次関数で補間してそれを解析積分する数値積分法

このプログラムは補間するための点の取り方とかを自動的に決めるためのアルゴリズムを含んでおり、積分範囲と被積分関数名を与えるだけで積分値を返す。

コード自身は外村氏のコードからとり、アルゴリズムを理解しようとしたが良く分からなかった。

```
      XIP=0.
      IF(A.EQ.B) GO TO 1009
      AA=MIN(A,B)
      BB=MAX(A,B)
      FAFB=FUNC(A)+FUNC(B)
      XH=BB-AA
      XJ=XH*.5*FAFB
      XNEW=AA+XH*.5
      XHA=XH/6.
      INDEX=0
1006  FNEWX=FUNC(XNEW)
      IF(INDEX.GT.0) GO TO 1003
      INDEX=1
      XI=XHA*(FAFB+FNEWX*4.)
1004  XJ=(XJ+XI*3.)*.25
      INDEX=INDEX+1
      IF(INDEX.GT.MAXI) GO TO 1009
      XH=XH*.5
      XNEW=AA+XH*0.5
      S=0.
1005  IF(XNEW.LT.BB) GO TO 1006
      XIP=(XJ+XH*2.*S)/3.
      IF(ABS(XIP-XI).LE.ABS(E*XIP)) GO TO 1009
      XI=XIP
      GO TO 1004
1003  S=S+FNEWX
      XNEW=XNEW+XH
      GO TO 1005
1009  RKSIMP=XIP
      IF(A.GT.B) RKSIMP=-RKSIMP
      RETURN
      END
```

2.7.14. rkstop.f

**********	最低次近似でストッピングパワーを与える関数

```
      REAL FUNCTION RKSTOP(CHSQ,BESQ,ZOVA,EXTE)
*                                                                      *
*     Ionization energy loss for proton per unit path length,          *
*     the "stopping power" [MeV g^-1 cm^2]                             *
*                                                                      *
*     Input parameters                                                 *
*                                                                      *
      IMPLICIT NONE
      REAL CHSQ ! particle charge squared
      REAL BESQ ! particle beta squared
      REAL ZOVA ! medium effective Z/A
      REAL EXTE ! medium effective mean excitation energy in MeV
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      INCLUDE 'kcmcon.i'
      INCLUDE 'kcpcon.i'

      REAL FK
      PARAMETER (FK=4.*M_PI*AVOGAD*ERADI*ERADI*EMASS)

      RKSTOP = FK*CHSQ*ZOVA/BESQ
     &*(LOG(2.*EMASS*BESQ/(1.-BESQ)/EXTE)-BESQ)

      RETURN
      END
```

粒子の性質は電荷二乗と速度二乗

媒質の性質は原子番号質量数比と平均励起エネルギー

陽子に関してはこれより高次の項は十分小さくこれで十分。ICRU のすべての効果を含んだ計算と比べ 250MeV まで飛程では 0.15mm 以下の差しかない。

2.8. REAL*8 関数

2.8.1. dkaa.f

```
************************************************************************
      REAL*8 FUNCTION DKAA(PX1,PX2)
*                                                                      *
*     Gaussian sigma squared of projected scattering angle             *
*     by Highland's or Lynch-Dahl's formula.                           *
```

2 種類の積分物質厚から横方向の 1 軸へ射影した散乱角の分散を与える関数

```
*     Reference: Lynch and Darl, Nucl. Instr. Meth. B58(1991)6-10      *
*                                                                      *
      IMPLICIT NONE
      REAL*8 PX1 ! Integral(z^2/p^2/beta^2/X0)dX
      REAL*8 PX2 ! Integral(z^2/beta^2/X0)dX (Lynch-Dahl's formula)
                 ! Integral(1/X0)dX          (Highland's formula)
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************

      REAL*8 S2,EPS
      DATA S2/13.6D0/    ! Parameter S2 in Highland's formula [MeV/c]
                         ! which is to give the projection angle's sigma
      DATA EPS/0.038D0/ ! Parameter epsilon in Highland's formula

      DKAA = (S2*(1.D0+EPS*LOG(MAX(1.D-3,PX2))))**2*PX1

      RETURN
      END
```

有名なハイランドの近似式とリンチ=ダールの改良版とではログ補正項で使う物質厚の定義のみが異なる。物質厚の計算はこの関数を呼ぶ側で計算する。

2.8.2. dkbesi.f

```
************************************************************************
      REAL*8 FUNCTION DKBESI(K,X)
*                                                                      *
*     Modified Bessel function Ik(x)                                   *
*     I have no idea what this is doing. I just took from Dr. Tomura's *
*     BTIS code.                                                       *
*                                                                      *
      IMPLICIT NONE
      INTEGER K ! integer order
      REAL*8 X  ! argument
*                                                                      *
*     19970725 - N. Kanematsu (Mitsubishi Electric Corporation)        *
************************************************************************
```

改ベッセル関数 Ik(x)を与える。ここで k は整数。

ベッセル関数の定義や計算法は数学の本を見てください。

```
      INTEGER N,I
      REAL*8 R,T,W,XX,EPS
      PARAMETER (EPS=1.D-10)

      N = ABS(K)

      R = 1.D0
      DO I = 1, N
         R = R*X/DBLE(2*I)
      END DO
      XX = 0.25D0*X*X
      T = 1.D0
      W = T
      I = 0
      DO WHILE (ABS(T).GT.ABS(W*EPS))
         I = I+1
         T = T*XX/DBLE(I*(I+N))
         W = W+T
      END DO

      DKBESI = R*W

      RETURN
      END
```


2.9. ユーザールーチン(例)

2.9.1. ukinit.f

```
      SUBROUTINE UKINIT
      IMPLICIT NONE

      INCLUDE 'kcbeam.i'
      INCLUDE 'kcmodu.i'
      INCLUDE 'kcpati.i'
      INCLUDE 'kcpenc.i'
      INCLUDE 'kccoll.i'
```

メインプログラムで最初に呼ばれ、照射条件を与えるためのユーザールーチン

```
      INCLUDE 'kcvolu.i'
      INCLUDE 'kcwobb.i'

      INTEGER I1,I2,I3,IVOX
      REAL X1,X2,X3,R

      WRITE(6,'(A,$)') 'UKINIT> Initialization --- '

C     Initial Beam
      CHRGBM = 1.0        ! proton charge
      AMASBM = 938.27231 ! proton mass
      FLUEBM = 1.0        ! number of protons
      EKINBM = 150.       ! kinetic energy
      SEKBM  = 0.         ! energy spread
      XYZBM(1) = 0.       ! incident point
      XYZBM(2) = 0.       !
      XYZBM(3) = 320.     ! (height)
      UVWBM(1) = 0.       ! incident direction
      UVWBM(2) = 0.       !
      UVWBM(3) =-1.       !

C     Range modulation (none)
      IVOLMD = 0
      NSTPMD = 1
      THICMD(1) = 0.
      RATIMD(1) = 1.

C     Volume definitions
      NVOLVL = 0

C     Wobbler WOB1
      NVOLVL = NVOLVL+1
      NAMEVL(NVOLVL) = 'WOB1' ! volume name
      CTYPVL(NVOLVL) = 'WOBB' ! volume type
      ISUBVL(NVOLVL) = 0      ! subindex
```

ビームの定義

ここではアイソセンタを原点にビーム方向を-z にする座標系をとっている

飛程変調はなし

ワブラー

```
      IMATVL(NVOLVL) = 1       ! material
      ZPOSVL(NVOLVL) = 287.    ! height
      THICVL(NVOLVL) = 61.     ! thickness
      PHIVL(NVOLVL)  = 0.      ! rotation about z
      IVOLWB = NVOLVL
      ANGLWB = 0.025

C     Scatterer SCT1
      NVOLVL = NVOLVL+1
      NAMEVL(NVOLVL) = 'SCT1'  ! volume name
      CTYPVL(NVOLVL) = 'DEGR'  ! volume type
      ISUBVL(NVOLVL) = 0       ! subindex
      IMATVL(NVOLVL) = 10      ! material
      ZPOSVL(NVOLVL) = 252.    ! height
      THICVL(NVOLVL) = 0.067   ! thickness
      PHIVL(NVOLVL)  = 0.      ! rotation about z

C     Degrader DEG1
      NVOLVL = NVOLVL+1
      NAMEVL(NVOLVL) = 'DEG1'  ! volume name
      CTYPVL(NVOLVL) = 'DEGR'  ! volume type
      ISUBVL(NVOLVL) = 0       ! subindex
      IMATVL(NVOLVL) = 3       ! material
      ZPOSVL(NVOLVL) = 200.    ! height
      THICVL(NVOLVL) = 2.      ! thickness
      PHIVL(NVOLVL)  = 0.      ! rotation about z

C     Pencil beams PEN1
      NVOLVL = NVOLVL+1
      NAMEVL(NVOLVL) = 'PEN1'  ! volume name
      CTYPVL(NVOLVL) = 'PENC'  ! volume type
      ISUBVL(NVOLVL) = 0       ! subindex
      IMATVL(NVOLVL) = 1       ! material
      ZPOSVL(NVOLVL) = 22.5    ! height
      THICVL(NVOLVL) = 0.      ! thickness
```

空気
中心高さ
厚さ

角度

散乱体

鉛
高さ
厚さ

ディグレーダ

ルサイト
高さ
厚さ

サブビーム生成面

空気
高さ

```
      PHIVL(NVOLVL)  = 0.      ! rotation about z
      IVOLPB = NVOLVL
      NXYPB(1) = 64
      NXYPB(2) = 64
      XY1PB(1) = -4.61
      XY1PB(2) = -4.61
      DXYPB(1) = 4.61*2.0/REAL(NXYPB(1))
      DXYPB(2) = 4.61*2.0/REAL(NXYPB(2))

C     Collimator COL1
      NVOLVL = NVOLVL+1
      NAMEVL(NVOLVL) = 'COL1' ! volume name
      CTYPVL(NVOLVL) = 'COLL' ! volume type
      ISUBVL(NVOLVL) = 0      ! subindex
      IMATVL(NVOLVL) = 1      ! material
      ZPOSVL(NVOLVL) = 22.5   ! height
      THICVL(NVOLVL) = 0.     ! thickness
      PHIVL(NVOLVL)  = 0.     ! rotation about z
      IVOLCO = NVOLVL
      DO I1 = 1, 2
         NXYCO(I1) = NXYPB(I1)
         XY1CO(I1) = XY1PB(I1)
         DXYCO(I1) = DXYPB(I1)
      END DO
      DO I1 = 1, NXYCO(1)
         X1 = XY1CO(1)+DXYCO(1)*(REAL(I1)-0.5)
         DO I2 = 1, NXYCO(2)
            X2 = XY1CO(2)+DXYCO(2)*(REAL(I2)-0.5)
            R = SQRT(X1*X1+X2*X2)
            IF (R.LT.4.61) THEN
               OPENCO(I1,I2) = .TRUE.
            ELSE
               OPENCO(I1,I2) = .FALSE.
            END IF
         END DO
```

物体番号
分割数

端の座標

ピクセル幅

コリメータ

高さ

ビーム軸回りの回転角

ピクセルはサブビーム面と同一とする

ピクセル中心の位置が半径 4.61cm 以下であればビームを通す

END DO	
C Patient PAT1	患者
NVOLVL = NVOLVL+1	
NAMEVL(NVOLVL) = 'PAT1' ! volume name	
CTYPVL(NVOLVL) = 'PATI' ! volume type	
ISUBVL(NVOLVL) = 0 ! subindex	
IMATVL(NVOLVL) = 2 ! material	水
ZPOSVL(NVOLVL) = 0. ! height	アイソセンタから高さが+-12.5cm の範囲を/KCVOLU/では患者領域とする
THICVL(NVOLVL) = 25. ! thickness	
PHIVL(NVOLVL) = 0. ! rotation about z	
IVOLPT = NVOLVL	
XYZOPT(1) = 0.	ビーム座標系で(0.,0.,2.6)の位置に患者座標系の原点を置く
XYZOPT(2) = 0.	
XYZOPT(3) = 2.6	
NXYZPT(1) = 100	ボクセルは 100×100×100 分割
NXYZPT(2) = 100	
NXYZPT(3) = 100	
XYZ1PT(1) = -10.	角の座標は患者座標系で(-10.,-10.,-10.)
XYZ1PT(2) = -10.	
XYZ1PT(3) = -10.	
DXYZPT(1) = 0.2	ボクセルサイズは 2mm 立方
DXYZPT(2) = 0.2	
DXYZPT(3) = 0.2	
IVOX = 0	各ボクセルに密度を与える
DO I3 = 1, NXYZPT(3)	
X3 = XYZ1PT(3)+DXYZPT(3)*(REAL(I3)-0.5)	
DO I2 = 1, NXYZPT(2)	
X2 = XYZ1PT(2)+DXYZPT(2)*(REAL(I2)-0.5)	
DO I1 = 1, NXYZPT(1)	
IVOX = IVOX+1	
X1 = XYZ1PT(1)+DXYZPT(1)*(REAL(I1)-0.5)	
R = SQRT(X1**2+X2**2)	
IF (X3.GE.7.0.AND.X3.LT.9.0.AND.R.GE.2.AND.R.LT.3.) THEN	ボクセル中心位置が患者座標系で 7cm から 9cm の高さ、半径で 2cm から 3cm に入ったらそのボクセルの密度は 2g/cm^{3}とする
DENSPT(IVOX) = 2.	

`` ELSE DENSPT(IVOX) = 1. END IF END DO END DO END DO WRITE(6,'(A)') 'done.' RETURN END ```	そうでなければ 1g/cm^3

2.9.2. uklast.f

`` SUBROUTINE UKLAST IMPLICIT NONE INCLUDE 'kcpati.i' INTEGER IKPVOX INTEGER NT,ND PARAMETER (NT=50,ND=100) REAL TMAX,DMAX,TMIN,DMIN PARAMETER (TMIN=0.,DMIN=0.) PARAMETER (TMAX=10.,DMAX=20.0) REAL DOSE(NT,ND),DERR(NT,ND),PQR(3),D,T,XYZ(3),P,Q,R EQUIVALENCE (P,PQR(1)),(Q,PQR(2)),(R,PQR(3)) INTEGER IP,IQ,IR,IVOX,IT,ID,I,J,NENT(NT,ND) INTEGER NH PARAMETER (NH=40000) REAL H(NH) COMMON/PAWC/H ```	メインプログラムで最後に呼ばれ、得られた線量分布を処理するユーザールーチン HBOOK のためのメモリー領域の確保

```
 WRITE(6,'(A,$)') 'UKLAST> Analysis --- '
 CALL FLUSH(6)

 CALL HLIMIT(NH)
 CALL HBOOK2(1,'Isodose; Radius (cm); Depth (cm)'
&,NT,TMIN,TMAX,ND,DMIN,DMAX,0.)

 CALL HBOOK2(2,'Isodose Error; Radius (cm); Depth (cm)'
&,NT,TMIN,TMAX,ND,DMIN,DMAX,0.)

 DO ID = 1, ND
    DO IT = 1, NT
       NENT(IT,ID) = 0
       DOSE(IT,ID) = 0.
       DERR(IT,ID) = 0.
    END DO
 END DO

 DO IR = 1, NXYZPT(3)
    R = XYZ1PT(3)+DXYZPT(3)*(REAL(IR)-0.5)
    DO IQ = 1, NXYZPT(2)
       Q = XYZ1PT(2)+DXYZPT(2)*(REAL(IQ)-0.5)
       DO IP = 1, NXYZPT(1)
          P = XYZ1PT(1)+DXYZPT(1)*(REAL(IP)-0.5)
          IVOX = IKPVOX(IP,IQ,IR)
          DO I = 1, 3
              XYZ(I) = XYZ0PT(I)
              DO J = 1, 3
                XYZ(I) = XYZ(I)+RP2VPT(I,J)*PQR(J)
             END DO
          END DO
          D = 12.6-XYZ(3)
          T = SQRT(XYZ(1)**2+XYZ(2)**2)
          IT = INT((T-TMIN)/(TMAX-TMIN)*REAL(NT)+1.)
```

等線量プロット用の二次元ヒストグラムの定義

配列の初期化

各ボクセルを見ていく

ビーム座標系でのボクセルの位置

体表面からの深さ
ビーム中心軸からの半径
深さビン番号

```
            ID = INT((D-DMIN)/(DMAX-DMIN)*REAL(ND)+1.)
            IF (IT.GE.1.AND.IT.LE.NT
     &         .AND.ID.GE.1.AND.ID.LE.ND) THEN
               NENT(IT,ID) = NENT(IT,ID)+1
               DOSE(IT,ID) = DOSE(IT,ID)+DOSEPT(IVOX)
               DERR(IT,ID) = DERR(IT,ID)+DOSEPT(IVOX)**2
            END IF
         END DO
      END DO
   END DO

   DO ID = 1, ND
      D = DMIN+(DMAX-DMIN)/REAL(ND)*(REAL(ID)-0.5)
      DO IT = 1, NT
         T = TMIN+(TMAX-TMIN)/REAL(NT)*(REAL(IT)-0.5)
         IF (NENT(IT,ID).GT.0) THEN
            DOSE(IT,ID) = DOSE(IT,ID)/REAL(NENT(IT,ID))
            DERR(IT,ID) = SQRT(MAX(0.,DERR(IT,ID)/REAL(NENT(IT,ID))
     &         -DOSE(IT,ID)**2))
         END IF
         CALL HFILL(1,T,D,DOSE(IT,ID))
         CALL HFILL(2,T,D,DERR(IT,ID))
      END DO
   END DO

   CALL HRPUT(0,'wobbler.hbook','NT')

   WRITE(6,'(A)') 'done.'
   CALL FLUSH(6)

   RETURN
   END
```

半径ビン番号

カウンターを加算
線量を加算
線量の二乗を加算

深さビン番号でループ

半径ビン番号でループ

このビンの平均線量
このビンの線量の標準偏差

平均線量をプロットに入れる
標準偏差をプロットに入れる

ファイルに保存する

終わり