Informes Técnicos Ciemat

1134
Febrero, 2008

TRUBA User Manual

M. A. Tereshchenko
F. Castejón
A. Cappa

Asociación EURATOM / CIEMAT para Fusión - 108

Laboratorio Nacional de Fusión por Confinamiento Magnético

Toda correspondencia en relación con este trabajo debe dirigirse al Servicio de Información y Documentación, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Ciudad Universitaria, 28040-MADRID, ESPAÑA.

Las solicitudes de ejemplares deben dirigirse a este mismo Servicio.

Los descriptores se han seleccionado del Thesauro del DOE para describir las materias que contiene este informe con vistas a su recuperación. La catalogación se ha hecho utilizando el documento DOE/TIC-4602 (Rev. 1) Descriptive Cataloguing On-Line, y la clasificación de acuerdo con el documento DOE/TIC.4584-R7 Subject Categories and Scope publicados por el Office of Scientific and Technical Information del Departamento de Energía de los Estados Unidos.

Catálogo general de publicaciones oficiales
http://www.060.es

Depósito Legal: M -14226-1995
ISSN: 1135 - 9420
NIPO: 654-08-010-6

Editorial CIEMAT

CLASIFICACIÓN DOE Y DESCRIPTORES

S70
PLASMA HEATING; BERNSTEIN MODE, ELECTRON PLASMA WAVES; TOKAMAK DEVICES; STELLARATORS; T CODES

TRUBA User Manual

Tereshchencko, M. A.[(1)]; Castejon, F.[(2)]; Cappa, A.[(2)]

24 pp. 2 figs. 13 refs.

Abstract:
The TRUBÁ (pipeline in Russian) code is a computational tool for studying thepropagation of Gaussian-shaped microwave beams in a prescribed equilibrium plasma. This manual covers the basic material hended to use the implementation of TRUBA (version 3.4) interfaced with the numerical library of the TJ-II stellarator. The manual provides a concise theoretical background of the problem, specifications for setting up the input files and interpreting the output of the code, and some information useful in modifying TRUBA.

Manual de Usuario de TRUBA

Tereshchencko, M. A.[(1)]; Castejon, F.[(2)]; Cappa, A.[(2)]

24 pp. 2 figs. 13 refs.

Resumen:
El código TRUBÁ (tubería en ruso) es una herramienta computacional para estudiar la propagación de haces gaussianos de mircroondas en un equilibrio de plasma prescrito. El presente manual muestra los materiales mínimos necesarios para usar la versión 3.2 del programa TRUBA, interconectada con la librería que provee las características del equilibrio del stellarator TJ-II. El manual aporta también un repaso teórico conciso al problema, las especificaciones de los ficheros de entrada y de salida del código y la información necesaria para las posibles futuras modificaciones de TRUBA.

1) Prokhorov Institute of General Physics, Russian Academy of Sciences, 119991 Moscow, Russia
2) Laboratorio Nacional de Fusión por Confinamiento Magnético, Asociación EURATOM-CIEMAT para Fusión, 28040 Madrid, Spain

Contents

Gaussian microwave beam in the complex eikonal form

In what follows, the wave field in a weakly inhomogeneous and stationary medium is sought in the form of a monochromatic Gaussian shaped beam

$$\mathbf{E}=\mathbf{E_0}(s)\exp\left\{i\frac{\omega}{c}\left(N_\alpha(s)\delta r_\alpha+\frac{1}{2}\Big(M_{\alpha\beta}(s)+iL_{\alpha\beta}(s)\Big)\delta r_\alpha\delta r_\beta\right)-i\omega t\right\}. \tag{1}$$

Hereinafter the Einstein sum convention for repeated indices is implied, and the coordinate system $\{r_\alpha\}$ is assumed to be Cartesian. In the above notation $\delta r_\alpha = r_\alpha - R_\alpha(s)$, where $\mathbf{R}(s)$ is the space curve of the central (reference) ray of the beam, with s being a parameter of this ray. The treatment (1) is also known as the paraxial WKB expansion of the wave field phase.

Matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ in (1) possess several properties, worthy of being noted. Evidently, they are symmetric:

$$M_{\alpha\beta}=M_{\beta\alpha}\,,\quad L_{\alpha\beta}=L_{\beta\alpha}\,. \tag{2}$$

Then, the vector $G_\alpha = N_\alpha(s)+\Big(M_{\alpha\beta}(s)+iL_{\alpha\beta}(s)\Big)\delta r_\beta$, which is the gradient of the complex eikonal, is necessarily a function of the position $\mathbf{r}$. Hence,

$$\frac{d}{ds}[\mathbf{G}]_{\mathbf{r}=\mathbf{R}(s)}=\left[\left(\frac{d\mathbf{R}}{ds}\cdot\nabla\right)\mathbf{G}\right]_{\mathbf{r}=\mathbf{R}(s)} \tag{3}$$

and thus we arrive at the additional constraints:

$$M_{\alpha\beta}\frac{dR_\beta}{ds}=\frac{dN_\alpha}{ds}\,,\quad L_{\alpha\beta}\frac{dR_\beta}{ds}=0\,. \tag{4}$$

Power transfer along the ray trajectory

The slowly varying amplitude $\mathbf{E_0}$ of the wave field (1), for the mode m under consideration, can be decomposed into a continuous superposition of plane waves by

$$\mathbf{E_0}(\mathbf{r})=\int\tilde{A}(\mathbf{N}+\delta\mathbf{n})\,\mathbf{e}^{(m)}(\mathbf{N}+\delta\mathbf{n})\exp\left(i\frac{\omega}{c}\delta\mathbf{n}\cdot\mathbf{r}\right)d^3\delta\mathrm{n}\,, \tag{5}$$

where $\tilde{A}$ is the spectral density, $\mathbf{e}^{(m)}$ is the unit eigenvector corresponding to the eigenvalue $\lambda^{(m)}$ of the dispersion tensor $\varLambda_{\alpha\beta}=n^2 I_{\alpha\beta}-n_\alpha n_\beta - K_{\alpha\beta}$, $\mathbf{n}$ is the normalized wave vector, $\boldsymbol{I}$ is the unit dyadic and $\boldsymbol{K}$ is the dielectric tensor. The dispersion relation for the given mode is $\lambda^{(m)}=0$. As shown in [1], for the case of narrow spectrum, the total wave power flux is

$$\mathbf{P}=\frac{c}{16\pi}|A|^2\left[\frac{\partial}{\partial\mathbf{n}}\left(\Lambda^{\mathrm{H}}_{\alpha\beta}e^{(m)*}_{\alpha}e^{(m)}_{\beta}\right)\right]_{\mathbf{n}=\mathbf{N}(s)} \tag{6}$$

and the sink term in the stationary power balance equation $\nabla\cdot\mathbf{P}=w$ is given by

$$w=-i\frac{\omega}{8\pi}|A|^2\left[\Lambda^{\mathrm{A}}_{\alpha\beta}e^{(m)*}_{\alpha}e^{(m)}_{\beta}\right]_{\mathbf{n}=\mathbf{N}(s)}, \tag{7}$$

with $A(s)=\int\tilde{A}(\mathbf{N}+\delta\mathbf{n})\exp(i\frac{\omega}{c}\delta\mathbf{n}\cdot\mathbf{R})d^3\delta\mathrm{n}$. Superscripts H and A denote Hermitian and anti-Hermitian parts of the tensor, respectively. Evidently, $\Lambda^{\mathrm{H}}_{\alpha\beta}e^{(m)*}_{\alpha}e^{(m)}_{\beta}=\mathrm{Re}\,\lambda^{(m)}$ and $\Lambda^{\mathrm{A}}_{\alpha\beta}e^{(m)*}_{\alpha}e^{(m)}_{\beta}=i\,\mathrm{Im}\,\lambda^{(m)}$. We posit that the relation $\mathrm{Re}\,\lambda^{(m)}=0$ is conserved along the reference ray hypercurve in the 6D phase space. Within the ansatz of (1), the 3D trajectory $\mathbf{r}=\mathbf{R}(s)$ is uniquely determined by the direction of $\mathbf{P}$. So it is easily seen that the tangent to this trajectory, i.e. the $d\mathbf{R}/ds$, is directed along the $[\partial\mathcal{H}/\partial\mathbf{n}]_{\mathbf{n}=\mathbf{N}(s)}$, where $\mathcal{H}(\mathbf{r},\mathbf{n})=f\,\mathrm{Re}\,\lambda^{(m)}$ and $f=f(\mathbf{r},\mathbf{n})$ is an arbitrary non-vanishing real function. As a result, one can define

$$\mathcal{H}=\prod_{j=0,1,2}\mathrm{Re}\,\lambda^{(j)}=\det(\boldsymbol{\Lambda}^{\mathrm{H}}). \tag{8}$$

However, when more than one eigenvalue tends to zero, it is necessary to somehow isolate the root corresponding to the required mode.

The aforesaid power balance equation in the ray coordinates takes the form

$$\frac{dP}{ds}=\mathrm{sgn}\left(\mathbf{P}\cdot\frac{d\mathbf{R}}{ds}\right)\left|\frac{d\mathbf{R}}{ds}\right|w, \tag{9}$$

Therefore, choosing the s norm such that

$$\frac{dR_\alpha}{ds}=-\frac{\partial\mathcal{H}}{\partial n_\alpha}, \tag{10}$$

we obtain

$$\frac{dP}{ds}=-2\frac{\omega}{c}P[f\,\mathrm{Im}\,\lambda^{(m)}]_{\mathbf{n}=\mathbf{N}(s)}, \tag{11}$$

where as before, $f=\mathcal{H}/\mathrm{Re}\,\lambda^{(m)}$. The invariant $\mathcal{H}=0$ holds all along the reference ray, and the essential condition $d\mathcal{H}/ds=0$, together with (10), brings to

$$\frac{dN_\alpha}{ds}=\frac{\partial\mathcal{H}}{\partial r_\alpha}. \tag{12}$$

[1] M. D. Tokman, E. Westerhof, M. A. Gavrilova. *Wave power flux and ray-tracing in regions of resonant absorption.* Plasma Phys. Control. Fusion **42** (2000) 91.

Clearly, the partial derivatives in (10) and (12) are to be evaluated on the reference ray. Equations (10) and (12) are usually referred to as ray-tracing equations, and the $\mathcal{H}(\mathbf{r},\mathbf{n})$ function is known as the ray Hamiltonian.

Shape of the Gaussian beam

Making use of the complex coupling $\boldsymbol{Q}=\boldsymbol{M}+i\boldsymbol{L}$, constraints (4) combined with (10) and (12) give rise to the second invariant of the reference ray

$$\hat{D}_\alpha \mathcal{H}=0\,,\qquad \hat{D}_\alpha=\frac{\partial}{\partial r_\alpha}+Q_{\alpha\beta}\frac{\partial}{\partial n_\beta}\,. \tag{13}$$

We note here that the total derivative along the reference ray can be written as

$$\frac{d}{ds}=\frac{\partial}{\partial s}+\frac{dR_\alpha}{ds}\hat{D}_\alpha\,. \tag{14}$$

The condition $d\big(\hat{D}_\alpha\mathcal{H}\big)/ds=0$ then leads to the sought beam-shape matrix equation

$$\frac{dQ_{\alpha\beta}}{ds}=\hat{D}_\beta\hat{D}_\alpha\mathcal{H}\,, \tag{15}$$

which can be splitted to give

$$\frac{dM_{\alpha\beta}}{ds}=\frac{\partial^2\mathcal{H}}{\partial r_\alpha\partial r_\beta}+\frac{\partial^2\mathcal{H}}{\partial r_\beta\partial n_\gamma}M_{\alpha\gamma}+\frac{\partial^2\mathcal{H}}{\partial r_\alpha\partial n_\gamma}M_{\beta\gamma}+\frac{\partial^2\mathcal{H}}{\partial n_\gamma\partial n_\delta}\big(M_{\alpha\gamma}M_{\beta\delta}-L_{\alpha\gamma}L_{\beta\delta}\big), \tag{16}$$

$$\frac{dL_{\alpha\beta}}{ds}=\left(\frac{\partial^2\mathcal{H}}{\partial r_\alpha\partial n_\gamma}+\frac{\partial^2\mathcal{H}}{\partial n_\gamma\partial n_\delta}M_{\alpha\delta}\right)L_{\beta\gamma}+\left(\frac{\partial^2\mathcal{H}}{\partial r_\beta\partial n_\gamma}+\frac{\partial^2\mathcal{H}}{\partial n_\gamma\partial n_\delta}M_{\beta\delta}\right)L_{\alpha\gamma}\,. \tag{17}$$

Equations (10) – (12), (16) and (17) constitute the system of the so-called beam-tracing equations [2]. The aforementioned constraints appearing now as

$$\mathcal{H}=0\,,\qquad M_{\alpha\beta}\frac{\partial\mathcal{H}}{\partial n_\beta}=-\frac{\partial\mathcal{H}}{\partial r_\alpha}\,,\qquad L_{\alpha\beta}\frac{\partial\mathcal{H}}{\partial n_\beta}=0 \tag{18}$$

should be used when setting the boundary values at $s=0$, and later as a check of consistency. The system of equations (16) and (17) can be solved separately from (10) and (12), provided that the trajectory of the reference ray is pre-computed. So it is useful in practice to decompose the beam-tracing procedure into the reference ray tracing with the subsequent beam-shaping.

[2] E. Poli, G. V. Pereverzev, A. G. Peeters. *Paraxial Gaussian wave beam propagation in an anisotropic inhomogeneous plasma.* Phys. Plasmas **6** (1999) 5.

Non-relativistic ray Hamiltonian

We cite here the well-known (see, e.g. [3]) expression for the Maxwellian-hot-plasma dispersion tensor in a local coordinate system such that $\mathbf{B}=(0,0,B)$ and $\mathbf{n}=(n_\perp,0,n_\parallel)$:

$$\boldsymbol{\Lambda}=\begin{bmatrix} n_\parallel^2-1 & 0 & -n_\perp n_\parallel \\ 0 & n^2-1 & 0 \\ -n_\perp n_\parallel & 0 & n_\perp^2-1 \end{bmatrix}$$
$$-\sum_\sigma q_\sigma \zeta_{0\sigma}\sum_{\kappa=-\infty}^{\infty}\begin{bmatrix} \frac{\kappa^2}{\lambda_\sigma}Y_{\kappa\sigma}Z_{\kappa\sigma} & i\kappa Y'_{\kappa\sigma}Z_{\kappa\sigma} & -\frac{(\pm)_\sigma\kappa}{\sqrt{2\lambda_\sigma}}Y_{\kappa\sigma}Z'_{\kappa\sigma} \\ -i\kappa Y'_{\kappa\sigma}Z_{\kappa\sigma} & \left(\frac{\kappa^2}{\lambda_\sigma}Y_{\kappa\sigma}-2\lambda_\sigma Y'_{\kappa\sigma}\right)Z_{\kappa\sigma} & i(\pm)_\sigma\sqrt{\frac{\lambda_\sigma}{2}}Y'_{\kappa\sigma}Z'_{\kappa\sigma} \\ -\frac{(\pm)_\sigma\kappa}{\sqrt{2\lambda_\sigma}}Y_{\kappa\sigma}Z'_{\kappa\sigma} & -i(\pm)_\sigma\sqrt{\frac{\lambda_\sigma}{2}}Y'_{\kappa\sigma}Z'_{\kappa\sigma} & -Y_{\kappa\sigma}\zeta_{\kappa\sigma}Z'_{\kappa\sigma} \end{bmatrix}, \qquad (19)$$

where σ runs over the plasma species, $(\pm)_\sigma=\mathrm{sgn}(\omega_{c\sigma})$, $q_\sigma=\omega_{p\sigma}^2/\omega^2$, $\lambda_\sigma=n_\perp^2/u_\sigma\mu_\sigma$, $u_\sigma=\omega_{c\sigma}^2/\omega^2$, $\mu_\sigma=m_\sigma c^2/T_\sigma$, $\zeta_{\kappa\sigma}=\sqrt{\mu_\sigma/2}\,n_\parallel^{-1}(1-(\pm)_\sigma\kappa\sqrt{u_\sigma})$, $Y_{\kappa\sigma}=I_\kappa(\lambda_\sigma)\exp(-\lambda_\sigma)$, $Z_{\kappa\sigma}=Z(\zeta_{\kappa\sigma})$, I_κ is the κth-order modified Bessel function of the first kind, $Z(\zeta)=i\int_0^\infty \exp(i\zeta t-t^2/4)\,dt$ is the nonrelativistic plasma dispersion function, which can be represented for real arguments as follows:

$$Z(x)=\exp(-x^2)\left(i\sqrt{\pi}-2\int_0^x\exp(t^2)dt\right). \qquad (20)$$

The primed variables in (19) denote the derivatives of the corresponding functions with respect to their actual arguments.

Since the use of TRUBA is focused upon the EC frequency range at present, the ion contribution has been neglected in this version and the sum over κ is truncated to $|\kappa|\leqslant 10$. In order to avoid the $u_e=1$ singularity in the determinant of (19), persisted in the $\mu_e\to\infty$ limit, the ray Hamiltonian is defined as $\mathcal{H}=(u_e-1)\det(\boldsymbol{\Lambda}^{\mathrm{H}})$.

The cold-limit expression

$$\mathcal{H}=(n^2-1)\left[(n^2-1+2q_e)(1-u_e-q_e)+q_e u_e(1+n_\parallel^2)\right]+q_e^2(1-q_e), \qquad (21)$$

whenever it would be used, is replaced by the following single-root Hamiltonian:

$$\mathcal{H}^{(m=1,2)}=\pm\max(D,D_{min})(n_\perp^2+(a_2\mp D)/2a_1), \quad D=\sqrt{a_2^2-4a_1a_3}, \qquad (22)$$

[3] M. Brambilla. *Kinetic theory of plasma waves, homogeneous plasmas.* (Oxford Univ. Press, Oxford, 1998).

where

$$a_1 = 1 - u_e - q_e,$$
$$a_2 = 2q_e(1-q_e) - (1-n_\parallel^2)\big(u_e q_e + 2(1-u_e-q_e)\big), \qquad (23)$$
$$a_3 = (1-q_e)\big((1-n_\parallel^2 - q_e)^2 - (1-n_\parallel^2)^2 u_e\big)$$

are the coefficients of $n_\perp^4$, $n_\perp^2$, and $n_\perp^0$ in the polynomial (21), and the customized small parameter $D_{min} > 0$ prevents the Hamiltonian from degradation when the modes are nearly degenerate (at the vacuum-plasma interface).

Both O- and X- modes at densities under the O-mode cutoff are treated using the cold-plasma Hamiltonian (22). The general hot-plasma Hamiltonian derived from (19) is “switched on” for $q_e > \frac{1}{2}\big(1 + (1+\sqrt{u_e})(1-n_\parallel^2)\big)$ so that to provide O-X mode conversion, and for $n_\perp^2 > 3$ thus allowing for electron Bernstein waves.

The TRUBA code proceeds with the O-X conversion in the following *ad hoc* manner. If the reflection point is revealed along the ray trajectory of the incident O-mode, and the width of the further evanescent layer is small enough, the launching point for the transmitted ray is to be determined on moving from that point along the density gradient until the dispersion relation is fulfilled again. It was shown in [4] that the ray trajectory continued from this point is asymptotically equivalent to the limiting central trajectory of the transmitted part of the wave packet. The wave vector of the launching ray is equated with its value at the reflection point. The fraction of the transmitted power is calculated using the one-dimensional O-mode tunneling theory [5,6]:

$$\eta = \exp\left\{-\pi \frac{\omega}{c} \ell_q \sqrt{\sqrt{u_e}/2}\left[2\sqrt{u_e}\left(1 - n_\parallel/n_{opt}\right)^2 + n_\perp^2\right]\right\}, \qquad (24)$$

where ℓ_q is the density gradient length and $n_{opt}^2 = \sqrt{u_e}/(1+\sqrt{u_e})$, with all the parameters taken at the reflection point.

Weakly relativistic ray Hamiltonian

The weakly relativistic approximation is referred to as the $\mu_e \gg 1$ condition, which is valid in most ECRH experiments. We start from the expression (2) from Ref. [7] that uses the

[4] A. V. Timofeev. *Electromagnetic oscillations near the critical surface in a plasma: Methodological note.* Plasma Phys. Rep. **27** (2001) 922.

[5] E. Mjølhus. *Coupling to Z mode near critical angle.* J. Plasma Phys. **31** (1984) 7.

[6] A. A. Zharov. *Theory of the conversion of normal waves in a nonuniform magnetized plasma.* Sov. J. Plasma Phys. **10** (1984) 642.

[7] I. P. Shkarofsky. *Dielectric tensor in Vlasov plasmas near cyclotron harmonics.* Phys. Fluids **9** (1966) 561.

only extra assumption of $\lambda_e \ll \mu_e$, thus still allowing for the short-wavelength Bernstein waves, which typically possess $\lambda_e \lesssim 20$. As before, we disregard the motion of ions. Following the technique of [7], but not restricting to $\lambda_e \ll 1$, we obtain the following:

$$\boldsymbol{\Lambda} = \begin{bmatrix} n_\parallel^2 - 1 & 0 & -n_\perp n_\parallel \\ 0 & n^2 - 1 & 0 \\ -n_\perp n_\parallel & 0 & n_\perp^2 - 1 \end{bmatrix}$$
$$+ q_e \mu_e \sum_{\kappa=-\infty}^{\infty} \begin{bmatrix} \frac{\kappa^2}{\lambda_e} \mathcal{R}_{3/2}^{(0)} & i\kappa \frac{\partial}{\partial \lambda_e} \mathcal{R}_{3/2}^{(0)} & \frac{\kappa}{\sqrt{2\lambda_e}} \mathcal{R}_{5/2}^{(1)} \\ -i\kappa \frac{\partial}{\partial \lambda_e} \mathcal{R}_{3/2}^{(0)} & \frac{\kappa^2}{\lambda_e} \mathcal{R}_{3/2}^{(0)} - 2\lambda_e \frac{\partial}{\partial \lambda_e} \mathcal{R}_{5/2}^{(0)} & -i\sqrt{\frac{\lambda_e}{2}} \frac{\partial}{\partial \lambda_e} \mathcal{R}_{5/2}^{(1)} \\ \frac{\kappa}{\sqrt{2\lambda_e}} \mathcal{R}_{5/2}^{(1)} & i\sqrt{\frac{\lambda_e}{2}} \frac{\partial}{\partial \lambda_e} \mathcal{R}_{5/2}^{(1)} & \mathcal{R}_{5/2}^{(0)} + \frac{1}{2} \mathcal{R}_{7/2}^{(2)} \end{bmatrix}, \tag{25}$$

where $\mathcal{R}_\nu^{(j)} = \left[\left(2\sqrt{a} \frac{\partial}{\partial z} \right)^j \mathcal{R}_\nu(z, a, \lambda_e, |\kappa|) \right]_{\substack{z=\mu_e(1+\kappa\sqrt{u_e}) \\ a=\mu_e n_\parallel^2/2}}$, $j = 0, 1, \ldots$, and

$$\mathcal{R}_\nu(z, a, \lambda, \kappa) = -i \int_0^\infty \exp\left(izt - \frac{\lambda + at^2}{1 - it} \right) I_\kappa\left(\frac{\lambda}{1 - it} \right) \frac{dt}{(1 - it)^\nu} \tag{26}$$

is the Robinson's relativistic plasma dispersion function [8]. This function may be expanded via usual Shkarofsky functions $\mathcal{F}_\nu(z, a) = -i \int_0^\infty \exp\left(izt - \frac{at^2}{1 - it} \right) \frac{dt}{(1 - it)^\nu}$:

$$\mathcal{R}_\nu(z, a, \lambda, \kappa) = \frac{\lambda^\kappa}{2^\kappa \kappa!} \sum_{j=0}^{\infty} g_j \mathcal{F}_{\nu+\kappa+j}(z, a), \quad g_0 = 1, \quad g_j = -\frac{2\kappa + 2j - 1}{j(2\kappa + j)} \lambda g_{j-1} \ . \tag{27}$$

Keeping only the leading term in (27) will reduce the dispersion tensor (25) to the standard $\lambda_e \ll 1$ weakly relativistic expression. The so-called moderately relativistic approximation [9] corresponds to retaining some higher-order terms of this expansion.

Except for $\lambda \ll 1$, use of (27) is computationally expensive and suffers from subtraction errors. In order to speed up the rate of the Hamiltonian evaluation, the following quite accurate approximation is implemented in the code:

$$\begin{aligned} \mathcal{R}_\nu(z, a, \lambda, \kappa) \approx \exp(-\lambda) \Big[& \big(I_\kappa(\lambda) - C_\kappa(\lambda) \big) \mathcal{F}_{\nu+\kappa}(z, a) \\ & + C_\kappa(\lambda) \big((1+h) \mathcal{F}_\nu(z-h, a) - h \mathcal{F}_\nu(z+1+h, a) \big) \Big], \end{aligned} \tag{28}$$

where $C_\kappa(\lambda) = \frac{2\lambda}{2\kappa+1} \big(I_\kappa(\lambda) - I_{\kappa+1}(\lambda) \big)$ and $h = (\sqrt{2} - 1)/2$. In order to preserve the peculiar behavior of the fundamental harmonic ($\kappa = -1$) contributions to the combinations

[8] P. A. Robinson. *Relativistic plasma dispersion functions.* J. Math. Phys. **27** (1986) 1206.
[9] D. G. Swanson. *Exact and moderately relativistic plasma dispersion functions.* Plasma Phys. Control. Fusion **44** (2002) 1329.

$K_{11}+K_{22}-2iK_{12}$ and $K_{13}+iK_{23}$, the λ-derivatives of the Robinson's function in (25) should be computed as

$$\frac{\partial}{\partial\lambda}\mathcal{R}_\nu(\ldots,|\kappa|)=\frac{|\kappa|}{\lambda}\mathcal{R}_\nu(\ldots,|\kappa|)-\mathcal{R}_{\nu+1}(\ldots,|\kappa|)+\mathcal{R}_{\nu+1}(\ldots,|\kappa|+1)\,, \tag{29}$$

at least for the fundamental harmonic.

The Shkarofsky function has a valuable asymptotic representation, in terms of the Z function and its high-order derivatives, in a domain of large real a:

$$\begin{aligned}\mathcal{F}_\nu(z,a)=-h\Big[&Z(\psi)+\upsilon_3h^3Z^{(3)}(\psi)+\upsilon_4h^4Z^{(4)}(\psi)+\upsilon_5h^5Z^{(5)}(\psi)\\&+\tfrac{1}{2}\upsilon_3^2h^6Z^{(6)}(\psi)+\upsilon_3\upsilon_4h^7Z^{(7)}(\psi)+\tfrac{1}{6}\upsilon_3^3h^9Z^{(9)}(\psi)\Big]+O(a^{-5/2})\,,\end{aligned} \tag{30}$$

with $h=(4a+2\nu)^{-1/2}$, $\psi=h(z+\nu)$, and $\upsilon_j=a+\nu/j$. If z is real, (30) is also valid in regions $-z\gg 1+\nu$ and $z-a\gg 1$. A further useful property is that $\mathcal{F}_\nu(z,a)$ of half-integer index ν can be expressed in terms of the "classical" plasma dispersion function [10]:

$$\begin{aligned}\mathcal{F}_{1/2}(z,a)&=-\frac{i}{2(z-a)^{1/2}}\left[Z(a^{1/2}+i(z-a)^{1/2})+Z(-a^{1/2}+i(z-a)^{1/2})\right]\\\mathcal{F}_{3/2}(z,a)&=-\frac{1}{2a^{1/2}}\left[Z(a^{1/2}+i(z-a)^{1/2})-Z(-a^{1/2}+i(z-a)^{1/2})\right]\end{aligned}\,, \tag{31}$$

and for $\kappa\geq 0$

$$\mathcal{F}_{\kappa+5/2}(z,a)=\frac{1}{a}\left[1+(a-z)\mathcal{F}_{\kappa+1/2}(z,a)-(\kappa+\tfrac{1}{2})\mathcal{F}_{\kappa+3/2}(z,a)\right]. \tag{32}$$

Practically, however, only the $\kappa<3$ functions can be safely treated using this method due to numerical instability of the recursion (32). Moreover, in the case of $|a|\ll 1$ this formula should be completely rejected. Instead of (32), the following relations can be used here:

$$\mathcal{F}_\nu(z,a)=e^{-a}\sum_{j=0}^{\infty}\frac{a^j}{j!}\mathcal{F}_{\nu+j}(z-a,0)\,,\qquad \mathcal{F}_{\nu+1}(z,0)=\frac{1}{\nu}\left(1-z\mathcal{F}_\nu(z,0)\right). \tag{33}$$

Similar to the splitting technique (22) of the non-relativistic Hamiltonian, O- and X- modes in low-density plasma are to be traced using the linearized Hamiltonian

$$\mathcal{H}^{(m=1,2)}=\pm\max\left\{\left|(u_e-1)\left[\partial\det(\boldsymbol{\Lambda}^{\mathrm{H}})/\partial(n_\perp^2)\right]_{n_{\perp m}^2}\right|,D_{min}\right\}\left(n_\perp^2-n_{\perp m}^2\right)\,, \tag{34}$$

where $n_{\perp m}^2$ is the corresponding root (computed numerically) of the $\det(\boldsymbol{\Lambda}^{\mathrm{H}})=0$ equation while the other parameters (i.e. q_e, u_e, μ_e, and $n_\parallel$) are kept fixed at their values.

[10] V. Krivenski, A. Orefice. *Weakly relativistic dielectric tensor and dispersion functions of a Maxwellian plasma*. J. Plasma Phys. **30** (1983) 125.

Absorption of the ray

In calculating the damping coefficient we make use of the relation

$$\left[\mathrm{Im}\,\lambda^{(m)}\prod_{j\neq m}\mathrm{Re}\,\lambda^{(j)}\right]_{\mathrm{Re}\,\lambda^{(m)}=0}=\mathrm{Im}\left[\det(\boldsymbol{\Lambda})-\det(\boldsymbol{\Lambda}^{\mathrm{A}})\right]_{n_{\perp m}^2}\ , \tag{35}$$

where $n_{\perp m}^2$ is the numerically computed root (recall Eq.(34)), so that equation (11) has the following versatile form:

$$\frac{dP}{Pds}=-2\frac{\omega}{c}\,\frac{\left[\partial\mathcal{H}/\partial(n_\perp^2)\right]_{n_{\perp m}^2}}{\left[\partial\det(\boldsymbol{\Lambda}^{\mathrm{H}})/\partial(n_\perp^2)\right]_{n_{\perp m}^2}}\,\mathrm{Im}\left[\det(\boldsymbol{\Lambda})-\det(\boldsymbol{\Lambda}^{\mathrm{A}})\right]_{n_{\perp m}^2}\ . \tag{36}$$

Reduction of L to the diagonal form

The objective is to find the transformation such that

$$\frac{\omega}{c} L_{\alpha\beta}\delta r_\alpha \delta r_\beta = \frac{w_1^2}{\rho_1^2} + \frac{w_2^2}{\rho_2^2}, \tag{37}$$

with ρ_i being the principal radii of the cross-sectional attenuation. Let $\mathbf{e}_3 = \frac{d\mathbf{R}}{ds}\left|\frac{d\mathbf{R}}{ds}\right|^{-1}$, then if $\mathbf{e}_3 \neq \nabla z$ we can introduce a local Cartesian coordinate system w_α with basis vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ lying in the cross-section of the beam, as shown in Fig. 1. Both systems are related via $\delta r_\alpha = T_{\alpha\beta} w_\beta$, with the generating matrix

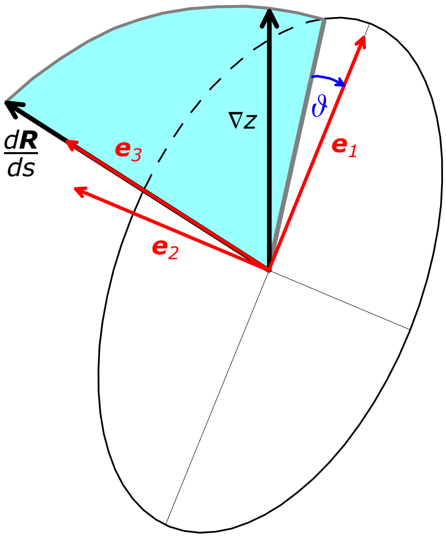

Figure 1

Attenuation ellipse in the cross-section of the beam and the associated orthonormal vector basis (red).

$$\boldsymbol{T} = \begin{bmatrix} \dfrac{-s_2\sin\vartheta - s_1 s_3\cos\vartheta}{s_\perp} & \dfrac{s_2\cos\vartheta - s_1 s_3 \sin\vartheta}{s_\perp} & s_1 \\ \dfrac{s_1\sin\vartheta - s_2 s_3\cos\vartheta}{s_\perp} & \dfrac{-s_1\cos\vartheta - s_2 s_3\sin\vartheta}{s_\perp} & s_2 \\ s_\perp\cos\vartheta & s_\perp \sin\vartheta & s_3 \end{bmatrix}, \tag{38}$$

where s_α are the laboratory coordinates of $\mathbf{e}_3$, $s_\perp = \sqrt{1-s_3^2}$ and ϑ is the rotation angle ensuring the diagonal form of $\boldsymbol{\mathcal{L}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{T}$. This angle or, rather, its sine and cosine should be found from the equation $\mathcal{L}_{12} = L_{\alpha\beta}T_{\alpha 1}T_{\beta 2} = 0$ which gives $\tan 2\vartheta = 2\chi$,

$$\chi = \frac{s_3\left[s_1 s_2\left(L_{11}-L_{22}\right) - \left(s_1^2 - s_2^2\right)L_{12}\right] + s_\perp^2\left(s_1 L_{23} - s_2 L_{13}\right)}{\left(s_1^2 s_3^2 - s_2^2\right)L_{11} + \left(s_2^2 s_3^2 - s_1^2\right)L_{22} + 2 s_1 s_2\left(1+s_3^2\right)L_{12} + s_\perp^4 L_{33} - 2 s_3 s_\perp^2\left(s_1 L_{13} + s_2 L_{23}\right)}, \tag{39}$$

and, hence,

$$\sin\vartheta = \frac{\operatorname{sgn}(\chi)}{\sqrt{2}}\sqrt{1 - 1/\sqrt{4\chi^2+1}}, \qquad \cos\vartheta = \frac{1}{\sqrt{2}}\sqrt{1+1/\sqrt{4\chi^2+1}}. \tag{40}$$

Having determined $\boldsymbol{T}$, we easily obtain $\mathbf{e}_i$ and $\rho_i = \left(\frac{\omega}{c}\mathcal{L}_{ii}\right)^{-1/2}$, $i = 1,2$. On restoring the laboratory representation, for given values of ϑ and ρ_i, the inverse transformation is straightforward ($\boldsymbol{T}$ is an orthogonal matrix): $\boldsymbol{L} = \boldsymbol{T}\boldsymbol{\mathcal{L}}\boldsymbol{T}^{\mathrm{T}}$.

Reduction of M to the diagonal form in vacuum

In plasma $\boldsymbol{M}$ is generally a nondegenerate matrix. This makes the diagonalization of $\boldsymbol{M}$ useless. On the contrary, in vacuum the first of the constraints (4) is reduced to $M_{\alpha\beta} N_\beta = 0$ so that the canonical transformation must give

$$N_\alpha \delta r_\alpha + \frac{1}{2} M_{\alpha\beta} \delta r_\alpha \delta r_\beta = w_3 + \frac{1}{2}\left(\frac{w_1^2}{p_1} + \frac{w_2^2}{p_2} \right), \tag{41}$$

with p_i being the principal focal parameters of the wave front paraboloid (they are equal to the principal radii of the wave front curvature on the beam axis). In (41) use is made of $N = 1$. The procedure of finding the diagonal form $\boldsymbol{\mathcal{M}} = \boldsymbol{T}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{T}$ is the same as for $\boldsymbol{L}$, with $\mathbf{e}_3 = \mathbf{N}$. The inverse focal parameters, which are of practical use, are given by $p_i^{-1} = \mathcal{M}_{ii}$.

Vacuum solution for the beam shape

In any homogeneous medium

$$\frac{dN_\alpha}{ds} = 0, \quad \frac{d}{ds}\left(M_{\alpha\beta} + iL_{\alpha\beta} \right) = \frac{\partial^2 \mathcal{H}}{\partial n_\gamma \partial n_\delta} \left(M_{\alpha\gamma} + iL_{\alpha\gamma} \right)\left(M_{\beta\delta} + iL_{\beta\delta} \right). \tag{42}$$

If the medium is also isotropic, as is the case with vacuum, then $\mathcal{H} = \mathcal{H}(n^2)$ so that

$$\frac{dR_\alpha}{ds} = -2N_\alpha \frac{\partial \mathcal{H}}{\partial n^2}, \quad \frac{\partial^2 \mathcal{H}}{\partial n_\alpha \partial n_\beta} = 2I_{\alpha\beta} \frac{\partial \mathcal{H}}{\partial n^2} + 4N_\alpha N_\beta \frac{\partial^2 \mathcal{H}}{\partial (n^2)^2}, \tag{43}$$

with $\boldsymbol{I}$ being the unit dyadic. Let us transform the coordinate system, for example using the same generating matrix (38) as employed above but with $\vartheta = 0$ and $\mathbf{e}_3 = \mathbf{N}$,

$$\boldsymbol{T} = \begin{bmatrix} \frac{-N_1 N_3}{s_\perp} & \frac{N_2}{s_\perp} & N_1 \\ \frac{-N_2 N_3}{s_\perp} & \frac{-N_1}{s_\perp} & N_2 \\ s_\perp & 0 & N_3 \end{bmatrix}, \quad s_\perp = \sqrt{1 - N_3^2}. \tag{44}$$

Now the non-vanishing beam coefficients are governed by the matrix Riccati equation

$$\frac{dQ_{ij}}{dw} = -Q_{ik} Q_{jk}, \tag{45}$$

where Latin indices range from 1 to 2, $\boldsymbol{Q} = \boldsymbol{T}^{\mathrm{T}}\left(\boldsymbol{M} + i\boldsymbol{L}\right)\boldsymbol{T}$, and the subscript 3 has been omitted from w. The analytic solution of (45) is

$$Q_{ij}(w) = \left[Q_{ij}(w_0) + I_{ij}Q_0\delta w\right]\left(1 + \left(Q_{11}(w_0) + Q_{22}(w_0) + Q_0\delta w\right)\delta w\right)^{-1}, \qquad (46)$$

where $Q_0 = Q_{11}(w_0)Q_{22}(w_0) - Q_{12}^2(w_0)$ and $\delta w = w - w_0$. Then, at the vacuum-plasma interface, one should restore the original coordinate representation $\boldsymbol{M} + i\boldsymbol{L} = \boldsymbol{T}\boldsymbol{Q}\boldsymbol{T}^{\mathrm{T}}$ that will be passed to the numerical solver.

Input files for the code

TRUBA.INI

line #	input parameters and explanation
1	$\boldsymbol{R_x}$, $\boldsymbol{R_y}$, $\boldsymbol{R_z}$ – Cartesian coordinates in *cm* for the origin of the central ray; the coordinate system coincides with that of TJ-II Library. Type: real.
2	Control flag ***F1*** for interpreting of the next line; see below. Type: integer.
3	Vector $\vec{\boldsymbol{N}}$. If ***F1***=*0*: $\boldsymbol{N_x}$, $\boldsymbol{N_y}$, $\boldsymbol{N_z}$ – absolute components, manually adjusted or taken from somewhere else. If ***F1***=*1*: $\boldsymbol{N_x h}$, $\boldsymbol{N_y h}$, $\boldsymbol{N_z h}$ – relative components; the value of ***h*** will be computed by the code. If ***F1***=*2*: $\boldsymbol{N_{\perp 1} h}$, $\boldsymbol{N_{\perp 2} h}$, $\boldsymbol{N_{\parallel}}$, where $\boldsymbol{N_{\parallel}}$ is the parallel component of $\vec{\boldsymbol{N}}$, $\boldsymbol{N_{\perp 1}}$ and $\boldsymbol{N_{\perp 2}}$ are the perpendicular components directed along $\nabla\psi$ and $\vec{\boldsymbol{B}}\times\nabla\psi$ correspondingly; the value of ***h*** will be computed by the code. If ***F1***=*3* and $\lvert \boldsymbol{q_e}\text{-}1\rvert<0.001$ (critical layer): $\boldsymbol{h_1}$, $\boldsymbol{h_2}$, $\boldsymbol{h_3}$, and $\vec{\boldsymbol{N}}$ will be constructed so as to $\boldsymbol{N}_{\perp 1,2} = 0.001\,\boldsymbol{h}_{1,2}\big/\sqrt{\boldsymbol{h}_1^2+\boldsymbol{h}_2^2}$, $\boldsymbol{N}_{\parallel} = \mathrm{sgn}(\boldsymbol{h_3})\sqrt{\sqrt{\boldsymbol{u_e}}/(1+\sqrt{\boldsymbol{u_e}})}$.
4	***f*** – Microwave frequency in *GHz*. Type: real.
5	Mode ***m*** of the wave to be launched: ***m***=*1* – O-mode, ***m***=*2* – X-mode, ***m***=*0* – hot plasma mode. Type: integer.
6	$\boldsymbol{n_1}$, $\boldsymbol{n_2}$, $\boldsymbol{n_3}$ – Coefficients of density profile $\boldsymbol{n_e} = \boldsymbol{n_1}(1\text{-}\psi^{\boldsymbol{n_2}})^{\boldsymbol{n_3}}\times 10^{13}$ cm^{-3}. Type: real. Or alternatively, the key word "tabular:" (case sensitive; preceding characters are ignored) and then the name of the file containing tabulated density profile (leading and trailing blanks are ignored).
7	$\boldsymbol{t_1}$, $\boldsymbol{t_2}$, $\boldsymbol{t_3}$ – Coefficients of electron temperature profile $\boldsymbol{T_e} = \boldsymbol{t_1}(1\text{-}\psi^{\boldsymbol{t_2}})^{\boldsymbol{t_3}}$ *keV*. Type: real. Or alternatively, the key word "tabular:" (case sensitive; preceding characters are ignored) and then the name of the file containing tabulated electron temperature profile (leading and trailing blanks are ignored).
8	$\Delta\boldsymbol{s}$ – Increment of the ray parameter for output; is also the interval of ***s*** for the ODE solver to integrate over, during each step. Type: real.
9	Control flag ***F2*** for the calculation mode: ***F2***=*0* – ray-tracing, ***F2***=*1* or *2* – beam-tracing. ***F2***=*1* is valid only for vacuum launch. Type: integer.
10	If ***F2***=*0* or *1*: ϑ_L, ρ_1, ρ_2, where ϑ_L is the rotation angle in *degrees* which is defined in Fig.1 and associated text, ρ_1 and ρ_2 are the principal radii of the beam attenuation ellipse in *cm*. If ***F2***=*1*, these values are used to initialize $\boldsymbol{L}_{\alpha\beta}$. If ***F2***=*0*, these values are used to initialize a parallel bunch of 4 characteristic surrounding rays to be traced right after the central ray (if one of ρs is zero, corresponding two rays will be taken away; if both ρs are zeros, only central ray will be traced). If ***F2***=*2*: $\boldsymbol{L_{xx}}$, $\boldsymbol{L_{xy}}$, $\boldsymbol{L_{xz}}$, $\boldsymbol{L_{yy}}$, $\boldsymbol{L_{yz}}$, $\boldsymbol{L_{zz}}$, in cm^{-1}. Type: real.

11	If **F2**=*0*, this line is ignored. If **F2**=*1*: ϑ_M, $1/p_1$, $1/p_2$, where ϑ_M is the rotation angle in *degrees* which ensures the diagonal form of $M_{\alpha\beta}$, $1/p_1$ and $1/p_2$ are the inverse principal focal parameters of the wave front in cm^{-1}. If **F2**=*2*: M_{xx}, M_{xy}, M_{xz}, M_{yy}, M_{yz}, M_{zz}, in cm^{-1}. Type: real.
12	Control flag ***F3*** for the dielectric tensor model: ***F3***=*0* – non-relativistic, ***F3***=*1* – weakly relativistic, ***F3***=*11* – "lite" weakly relativistic (valid if $\lambda_e \ll 1$, up to the 2nd EC harmonic). Type: integer.
13	Control flag ***F4*** for the choice of ODE solver: ***F4***=*1* – D02EJF (NAG), ***F4***=*2* – LSODE/non-stiff (ODEPACK), ***F4***=*3* – LSODA (ODEPACK). Type: integer.
14	Relative tolerance parameter for ODE solvers. Type: real.
15	Control flag ***F5*** for the choice of the ray-tracing termination condition: ***F5***=*1* – $P(s)/P(0)<10^{-9}$, ***F5***=*2* – $N_\perp^2>10^3$ *and* $\|u_e\text{-}1\|<10^{-3}$, ***F5***=*3* – $N_\perp^2/u_e\mu_e>50$, ***F5***=*4* – any of the last two criteria fulfilled. By default, tracing goes on till the ray comes upon the wall of the vacuum vessel. Type: integer.
16	Control flag ***F6*** for the choice of output style of the RAY∗.DAT files: ***F6***=*1* – all the values in the lines #4 and #5 are preceded with its titles (see below), ***F6***=*0* – the same without the titles. Type: integer.
17	Control flag ***F7*** for the choice of output mode for the POWER∗.DAT files: ***F7***=*1* – two more columns are output if $m=1$ or 2 (see below), ***F7***=*0* – optional output is suppressed. Type: integer.
18	Control flag ***F8*** for the choice of output mode for the B_U_G_S file: ***F8***=*2* – detailed auxiliary/debugging information, ***F8***=*1* – only checkpoint data, ***F8***≤*0* – no output. If ***F8***=*-1*, all the output to the screen is suppressed, except for the error messages. Type: integer.

TJ2_B.INI

line #	input parameters and explanation
1	Name of the file containing the coefficients of a given configuration. To be passed to the initialization routine of the TJ-II Library. Type: character.
2	Name of the file containing the namelist with the currents flowing trough the different coils of the device. To be passed to the initialization routine of the TJ-II Library. Type: character.
3	Artificial scale factor for the magnetic field returned by the TJ-II Library. Type: real.

File indicated in the 6th line of TRUBA.INI, if any.

line #	input parameters and explanation	
1	***j_n*** – The number of tabulated density values (type: integer); $[dn_e/d\psi]_{\psi=0}$ and $[dn_e/d\psi]_{\psi=1}$ in 10^{13} cm^{-3} (type: real).	
2 — ***j_n***+1	Values of ψ, in the order of increasing from *0* to *1*. Type: real.	Corresponding values of density n_e in 10^{13} cm^{-3}. Type: real.

File indicated in the 7th line of TRUBA.INI, if any.

line #	input parameters and explanation	
1	***j_t*** – The number of tabulated electron temperature values (type: integer); $[dT_e/d\psi]_{\psi=0}$ and $[dT_e/d\psi]_{\psi=1}$ in keV (type: real).	
2 — ***j_t***+1	Values of ψ, in the order of increasing from *0* to *1*. Type: real.	Corresponding values of electron temperature T_e in keV. Type: real.

Files indicated in the first two lines of TJ2_B.INI.

For any details see the `init_tj2_lib` routine documentation.

Output of the code

Screen, the asterisk (∗) unit.

If ***F8***≠*-1*:

at the origin of the central ray: q= q_e, u= u_e, flux= ψ;

if ***F1***=*1*: » N is multiplied by h, if ***F1***=*2*: » Nt2 is set to $N_\perp^2$;

Starting ray tracing…;

while tracing: the number of step, m, $N_\perp^2$, at the end: ψ, q_e;

if ***F2***=*1*, *2*: Starting beam shaping…;

while shaping: the number of step.

Whenever:

diagnoses of abnormalities.

RAY0.DAT [RAY1.DAT, RAY2.DAT, etc.]*

line # in record block	output values and explanation	
1	In vacuum pre-tracing: *0*. During tracing in plasma: the number of step. In vacuum post-tracing: *-1*, except the first record after exit: *-2*. Also *-2* in the case of preconditioned stop in plasma. Type: integer.	In vacuum: *0*. In plasma: current value of $\lambda_e = N_\perp^2/u_e\mu_e$. Type: real.
2	R_x, R_y, R_z – Current coordinates in *cm* of the reference ray. Type: real.	
3	N_x, N_y, N_z – Components of the vector $\vec{N}$ on the reference ray. Type: real.	
4	Current values of $N_\perp^2, N_\parallel^2, q_e, u_e$ (preceded with the titles if ***F6***=*1*). In vacuum the first three are zeroes. Type: real.	
5	Current value of $H = \mathcal{H}/\max(1, N_\perp^4)$ (preceded with the title if ***F6***=*1*). In vacuum: *0*. Type: real.	

* The files, which are not necessarily output, will henceforth be colored blue.

RAY0T.DAT [RAY1T.DAT, RAY2T.DAT, etc.]

The same as above, for the rays underwent tunneling through the q_e=*1* layer (O-mode only). The first record block is for the point just before tunneling (found from minimum $N_\perp^2$ <*0.01* on the ray trajectory); the second block is for the first point of the transmitted ray. The 5th line in these two blocks contains $N_{\perp 2}^2$, instead of H. No more peculiarities. The step numbering continues the original one.

POWER0.DAT [POWER1.DAT, POWER2.DAT, etc.]

1st column	2nd column*	3rd column*	last column
The number of step.	Im($\boldsymbol{N}$)	$\exp\left(-2\frac{\omega}{c}\sum \mathrm{Im}(\boldsymbol{N})\Delta w\right)$	P/P_1, with $P_1=P$ in the first line.

* These columns are output only if $F7=1$ and $m=1$ or 2.

POWER0T.DAT [POWER1T.DAT, POWER2T.DAT, etc.]

The same as above, for the rays underwent tunneling through the $q_e=1$ layer (O-mode only). The only difference: the 3rd column and the last column are multiplied by the transmission coefficient (damping before the tunneling is out of account here).

BEAM0.DAT

line # in record block	output values and explanation
1	The number of step (or *0*, *-1*, *-2*), synchronized with the RAY0.DAT. Type: integer.
2	Coordinates of the $\vec{e}_1$ basis vector of the beam attenuation ellipse. Type: real.
3	Coordinates of the $\vec{e}_2$ basis vector of the beam attenuation ellipse. Type: real.
4	ρ_1, ρ_2 – Principal radii of the beam attenuation ellipse in *cm*. Type: real.
5	M_{xx}, M_{xy}, M_{xz}, M_{yy}, M_{yz}, M_{zz}, in cm^{-1}. Type: real.

REFLECTION

line # in record block	output values and explanation
1	R_x, R_y, R_z – Cartesian coordinates in *cm* of the emergent ray spot on the vessel.
2	N_x, N_y, N_z – Components of the vector $\vec{N}$ for the ray reflected from the vessel.

Such information on all rays traced is successively collected in this file.

Damp_profile0.dat [Damp_profile1.dat, Damp_profile2.dat, etc.]

1st column	2nd column
$\sqrt{\psi_i}$, i.e. the normalized effective radius; i = line number: from *1* to *200*.	$\dfrac{\delta W_{abs}(\psi_i^- \le \psi_i < \psi_i^+)/W_0}{\psi_i^+ - \psi_i^-}$, where $\psi_i^\pm = (\sqrt{\psi_i} \pm 1/400)^2$.

To calculate the net radial profile of power deposition per unit volume $d\boldsymbol{W_{abs}}/d\boldsymbol{V}$ from the results of a multiple ray-tracing simulation, one has to sum up the 2nd–column vectors for all the rays of a bunch (correspondingly weighted), then multiply the result by the total power of the beam and divide it by the plasma volume (or, more strictly, by $d\boldsymbol{V}/d\psi$).

Damp_profile0T.dat [Damp_profile1T.dat, Damp_profile2T.dat, etc.]

The same as above, for the rays underwent tunneling around the $\boldsymbol{q_e}=1$ layer (O-mode only). The transmission efficiency is taken into account here, except for a damping of the ray before the tunneling.

B_U_G_S

Intended for output of any data helpful in analysis of results or debugging (if ***F8***=*2*). If ***F8***=*1*, only the changes of the wave mode along the ray trajectory are registered, successively for all the rays traced.

Internal structure of the code

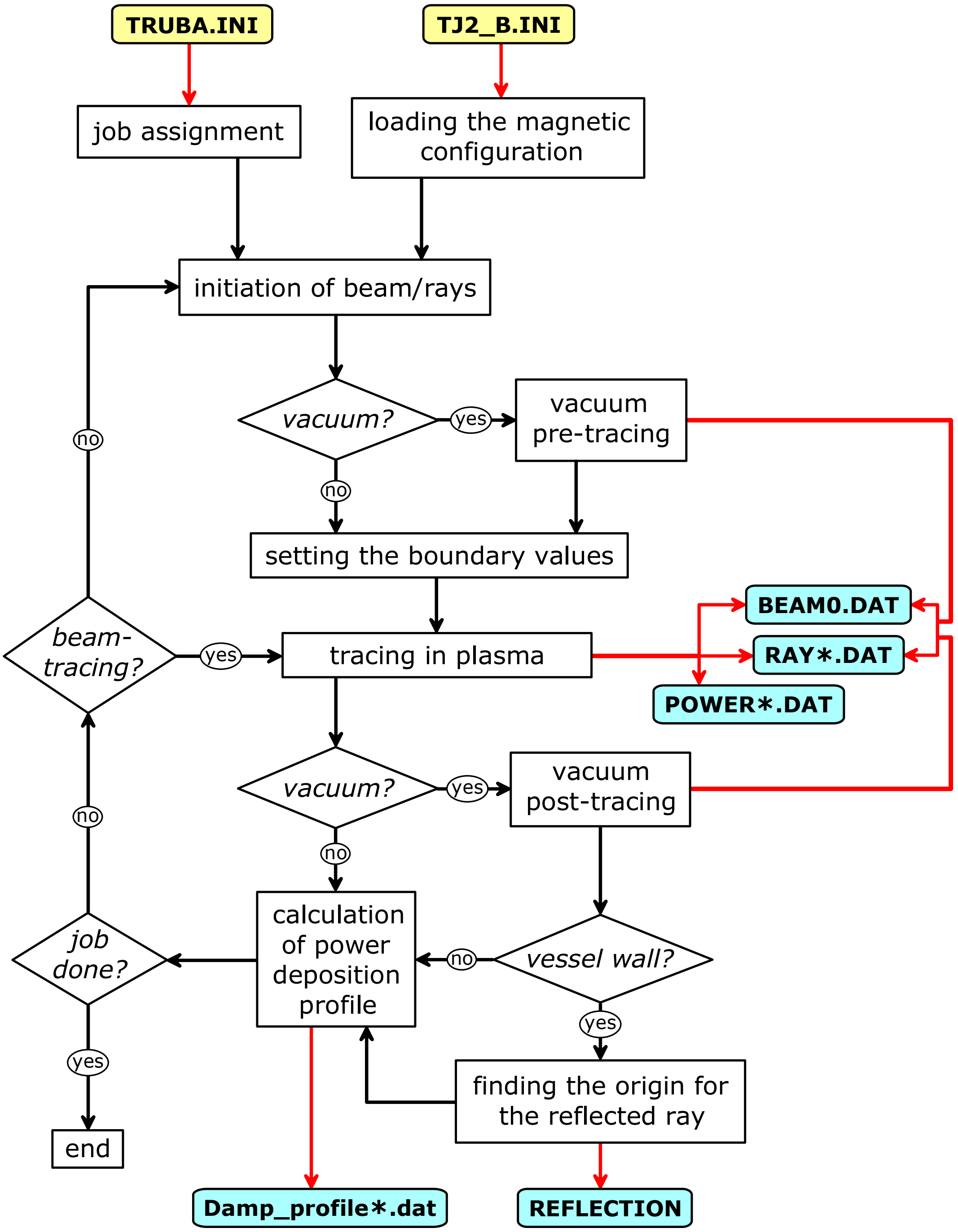

Figure 2
Flow chart of TRUBA, showing the principal steps involved in a run of the code.

Contents of the beam-data array $\boldsymbol{O}$(0:18)

array component	value, in beam- or ray-tracing mode	
$\boldsymbol{O}(0)$	$\boldsymbol{P}(\boldsymbol{s})/\boldsymbol{P}(0)$	
$\boldsymbol{O}(1)$	$\boldsymbol{R}_{x}(\boldsymbol{s})$	
$\boldsymbol{O}(2)$	$\boldsymbol{R}_{y}(\boldsymbol{s})$	
$\boldsymbol{O}(3)$	$\boldsymbol{R}_{z}(\boldsymbol{s})$	
$\boldsymbol{O}(4)$	$\boldsymbol{N}_{x}(\boldsymbol{s})$	
$\boldsymbol{O}(5)$	$\boldsymbol{N}_{y}(\boldsymbol{s})$	
$\boldsymbol{O}(6)$	$\boldsymbol{N}_{z}(\boldsymbol{s})$	
$\boldsymbol{O}(7)$	$\boldsymbol{M}_{xx}(\boldsymbol{s})$	$\boldsymbol{R}_{x}(0)$
$\boldsymbol{O}(8)$	$\boldsymbol{L}_{xx}(\boldsymbol{s})$	$\boldsymbol{R}_{y}(0)$
$\boldsymbol{O}(9)$	$\boldsymbol{M}_{xy}(\boldsymbol{s})$	$\boldsymbol{R}_{z}(0)$
$\boldsymbol{O}(10)$	$\boldsymbol{L}_{xy}(\boldsymbol{s})$	$\boldsymbol{N}_{x}(0)$
$\boldsymbol{O}(11)$	$\boldsymbol{M}_{xz}(\boldsymbol{s})$	$\boldsymbol{N}_{y}(0)$
$\boldsymbol{O}(12)$	$\boldsymbol{L}_{xz}(\boldsymbol{s})$	$\boldsymbol{N}_{z}(0)$
$\boldsymbol{O}(13)$	$\boldsymbol{M}_{yy}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{1}(0)\cdot\nabla\boldsymbol{x}$
$\boldsymbol{O}(14)$	$\boldsymbol{L}_{yy}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{1}(0)\cdot\nabla\boldsymbol{y}$
$\boldsymbol{O}(15)$	$\boldsymbol{M}_{yz}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{1}(0)\cdot\nabla\boldsymbol{z}$
$\boldsymbol{O}(16)$	$\boldsymbol{L}_{yz}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{2}(0)\cdot\nabla\boldsymbol{x}$
$\boldsymbol{O}(17)$	$\boldsymbol{M}_{zz}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{2}(0)\cdot\nabla\boldsymbol{y}$
$\boldsymbol{O}(18)$	$\boldsymbol{L}_{zz}(\boldsymbol{s})$	$\vec{\boldsymbol{e}}_{2}(0)\cdot\nabla\boldsymbol{z}$

Here $\vec{\boldsymbol{e}}_1$ and $\vec{\boldsymbol{e}}_2$ are the principal orts of the beam attenuation ellipse.

Synopsis of subroutines

SUBROUTINE Local_1(r(3), B(3), modB, q, u, t, fl)

Input: ***r***(1:3)=(***x***, ***y***, ***z***) – Cartesian coordinates of a point in *cm*.

Output: ***B***(1:3)=($\boldsymbol{B_x}$, $\boldsymbol{B_y}$, $\boldsymbol{B_z}$) – components of the magnetic field $\vec{\boldsymbol{B}}$ in *tesla*, $\boldsymbol{modB} = |\vec{\boldsymbol{B}}|$ in *tesla*, $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \boldsymbol{\mu_e^{-1}}$, ***fl*** – normalized magnetic flux.

SUBROUTINE Local_2(r(3), B(3), modB, q, u, t, Dq(3), Dt(3), DB(3,3))

Input: ***r***(1:3)=(***x***, ***y***, ***z***) – Cartesian coordinates of a point in *cm*.

Output: ***B***(3) – magnetic field $\vec{\boldsymbol{B}}$ in *tesla*, $\boldsymbol{modB} = |\vec{\boldsymbol{B}}|$ in *tesla*, $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \boldsymbol{\mu_e^{-1}}$, ***Dq***(3) – gradient of $\boldsymbol{q_e}$ in *cm*$^{-1}$, ***Dt***(3) – gradient of $\boldsymbol{\mu_e^{-1}}$ in *cm*$^{-1}$, ***DB***(3,3) – gradient of $\vec{\boldsymbol{B}}$ in *cm*$^{-1}$ ($\boldsymbol{DB}(\mathrm{i,j}) = \boldsymbol{dB_i/dr_j}$).

SUBROUTINE Local_3(r(3), B(3), modB, q, u, t, Dq(3), Dt(3), Du(3), DDq(3,3), DDt(3,3), DDu(3,3), Z2(3,3), Z3(3,3,3), Z4(3,3,3,3))

Input: ***r***(1:3)=(***x***, ***y***, ***z***) – Cartesian coordinates of a point in *cm*.

Output: $\boldsymbol{B}(\mathrm{i}) = \boldsymbol{B_i}$ in *tesla*, $\boldsymbol{modB} = \boldsymbol{B}$ in *tesla*, $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \boldsymbol{\mu_e^{-1}}$, $\boldsymbol{Dq}(\mathrm{i}) = \boldsymbol{dq_e/dr_i}$ in *cm*$^{-1}$, $\boldsymbol{Dt}(\mathrm{i}) = \boldsymbol{d\mu_e^{-1}/dr_i}$ in *cm*$^{-1}$, $\boldsymbol{Du}(\mathrm{i}) = \boldsymbol{du_e/dr_i}$ in *cm*$^{-1}$, $\boldsymbol{DDq}(\mathrm{i,j}) = \boldsymbol{d^2q_e/dr_idr_j}$ in *cm*$^{-2}$, $\boldsymbol{DDt}(\mathrm{i,j}) = \boldsymbol{d^2\mu_e^{-1}/dr_idr_j}$ in *cm*$^{-2}$, $\boldsymbol{DDu}(\mathrm{i,j}) = \boldsymbol{d^2u_e/dr_idr_j}$ in *cm*$^{-2}$, $\boldsymbol{Z2}(\mathrm{i,j}) = \boldsymbol{d(B_j/B)/dr_i}$ in *cm*$^{-1}$, $\boldsymbol{Z3}(\mathrm{i,j,k}) = \boldsymbol{d(B_jB_k/B^2)/dr_i}$ in *cm*$^{-1}$, $\boldsymbol{Z4}(\mathrm{i,j,k,l}) = \boldsymbol{d(B_l/B \times d(B_k/B)/dr_i)/dr_j}$ in *cm*$^{-2}$.

SUBROUTINE DIELECTR_NR(q, u, t, nl2, nt2, K(3,3))

Input: $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \boldsymbol{\mu_e^{-1}}$, $\boldsymbol{nl2} = \boldsymbol{n_\parallel^2}$, $\boldsymbol{nt2} = \boldsymbol{n_\perp^2}$.

Output: ***K***(3,3) – non-relativistic dielectric tensor, as it appears in (19).

SUBROUTINE DIELECTR_WR(q, u, t, nl2, nt2, K(3,3))

Input: $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \boldsymbol{\mu_e^{-1}}$, $\boldsymbol{nl2} = \boldsymbol{n_\parallel^2}$, $\boldsymbol{nt2} = \boldsymbol{n_\perp^2}$.

Output: ***K***(3,3) – weakly relativistic dielectric tensor, as it appears in (25), with the use of (28).

SUBROUTINE DIELECTR_WR1(q, u, t, nl2, nt2, K(3,3))

Input: $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \mu_{\boldsymbol{e}}^{\boldsymbol{-1}}$, $\boldsymbol{nl2} = \boldsymbol{n}_{\parallel}^{\boldsymbol{2}}$, $\boldsymbol{nt2} = \boldsymbol{n}_{\perp}^{\boldsymbol{2}}$.

Output: $\boldsymbol{K}(3,3)$ – weakly relativistic dielectric tensor, as it appears in (25), with the use of the expansion (27) retaining only those terms of lower orders that would be necessary and sufficient for frequencies up to the 2nd EC harmonic in the $\lambda_{\boldsymbol{e}} \ll 1$ limit.

SUBROUTINE ZERO_DTH(m, q, u, t, nl2, nt2, nt2m)

Input: mode $\boldsymbol{m}$ of the wave, $\boldsymbol{q} = \boldsymbol{q_e}$, $\boldsymbol{u} = \boldsymbol{u_e}$, $\boldsymbol{t} = \mu_{\boldsymbol{e}}^{\boldsymbol{-1}}$, $\boldsymbol{nl2} = \boldsymbol{n}_{\parallel}^{\boldsymbol{2}}$, $\boldsymbol{nt2} = \boldsymbol{n}_{\perp}^{\boldsymbol{2}}$.

Output: $\boldsymbol{nt2m} = \boldsymbol{n}_{\perp \boldsymbol{m}}^{\boldsymbol{2}}$ – the root of the dispersion equation:

… corresponding to the given mode, if $\{\boldsymbol{m} = 1$ or $2\}$ and $\{[\boldsymbol{u_e}>1$ and $\boldsymbol{q_e}<0.1]$ or $[\boldsymbol{u_e}<1$ and $\boldsymbol{q_e}<(1-\boldsymbol{n}_{\parallel}^{\boldsymbol{2}})(1-\sqrt{\boldsymbol{u_e}})]\}$;

… otherwise, closest to the $\boldsymbol{n}_{\perp}^{\boldsymbol{2}}$ input.

SUBROUTINE DAMP(O(6), g)

Input: $\boldsymbol{O}(1:6)$ containing $\boldsymbol{O}(1:3) = \vec{\boldsymbol{R}}$ in *cm*, and $\boldsymbol{O}(4:6) = \vec{\boldsymbol{N}}$.

Output: $\boldsymbol{g} = \lim\limits_{\mathcal{H}\to 0} \left\{ \mathrm{Im}\left(\det(\boldsymbol{\Lambda}) - \det(\boldsymbol{\Lambda}^{\mathrm{A}}) \right) \mathcal{H} / \det(\boldsymbol{\Lambda}^{\mathrm{H}}) \right\}$ – damping coefficient, so that (11) takes the form $\boldsymbol{dP/ds} = -2 \dfrac{\boldsymbol{\omega}}{\boldsymbol{c}} \boldsymbol{gP}$.

SUBROUTINE QDAMP(a, b, Oa(6), Ob(6), G)

Input: $\boldsymbol{a}$ and $\boldsymbol{b}$ – $\boldsymbol{s}$-coordinates of adjacent points of the ray trajectory; $\boldsymbol{Oa}(1:6)$ containing $\boldsymbol{Oa}(1:3) = \vec{\boldsymbol{R}}(\boldsymbol{a})$ in *cm*, $\boldsymbol{Oa}(4:6) = \vec{\boldsymbol{N}}(\boldsymbol{a})$; and $\boldsymbol{Ob}(1:6)$ containing $\vec{\boldsymbol{R}}(\boldsymbol{b})$, $\vec{\boldsymbol{N}}(\boldsymbol{b})$.

Output: $\boldsymbol{G}$ – integral of the damping coefficient $\boldsymbol{g}$ (see above), from $\boldsymbol{a}$ to $\boldsymbol{b}$, taken along the spline-interpolated hyper-trajectory in 6D phase space.

SUBROUTINE ZERO_F(F, a, b, Fa, Fb, c)

Input: $\boldsymbol{F}$ – user-supplied real function of which a zero will be found, between $\boldsymbol{a}$ and $\boldsymbol{b}$; $\boldsymbol{Fa} = \boldsymbol{F}(\boldsymbol{a})$ and $\boldsymbol{Fb} = \boldsymbol{F}(\boldsymbol{b})$ must be opposite in sign.

Output: $\boldsymbol{c}$ – computed zero.

SUBROUTINE D1H(O(6), O1(6))

Input: $\boldsymbol{O}(1:6)$ containing $\boldsymbol{O}(1:3) = \vec{\boldsymbol{R}}$ in *cm*, and $\boldsymbol{O}(4:6) = \vec{\boldsymbol{N}}$.

Output: $\boldsymbol{O1}(1:6)$ containing $\boldsymbol{O1}(1:3) = \partial\mathcal{H}/\partial\vec{\boldsymbol{r}}$ in *cm*$^{-1}$, and $\boldsymbol{O1}(4:6) = \partial\mathcal{H}/\partial\vec{\boldsymbol{n}}$.

SUBROUTINE D2HX(O(6), X(6,6))

Input: $\boldsymbol{O}(1{:}6)$ containing $\boldsymbol{O}(1{:}3)=\vec{\boldsymbol{R}}$ in *cm*, and $\boldsymbol{O}(4{:}6)=\vec{\boldsymbol{N}}$.

Output: $\boldsymbol{X}(1{:}6,1{:}3)=\partial[\boldsymbol{O1}(1:6)]/\partial\vec{\boldsymbol{r}}$ and $\boldsymbol{X}(1{:}6,4{:}6)=\partial[\boldsymbol{O1}(1:6)]/\partial\vec{\boldsymbol{n}}$ at given point; $\boldsymbol{O1}(1{:}6)$ is the output of D1H.

SUBROUTINE FRHS0(X, Y(6), DY(6))

Input: ***X*** – independent variable, ***Y***(6) – solution at ***X***.

Output: ***DY***(6) – RHS of the system of ODEs $\boldsymbol{dy_i/dx}=\boldsymbol{f_i}(\boldsymbol{x},\boldsymbol{y_1},\ldots,\boldsymbol{y_6})$, $\boldsymbol{i}=1,\ldots,6$, assembled of (10) and (12).

SUBROUTINE FRHS1(X, Y(12), DY(12))

Input: ***X*** – independent variable, ***Y***(12) – solution at ***X***.

Output: ***DY***(12) – RHS of the system of ODEs $\boldsymbol{dy_i/dx}=\boldsymbol{f_i}(\boldsymbol{x},\boldsymbol{y_1},\boldsymbol{y_2},\ldots,\boldsymbol{y_{12}})$, $\boldsymbol{i}=1,2,\ldots,12$, assembled of (16) and (17).

SUBROUTINE FRWRDT(e(3), X(6), T(3,3))

Input: ***e***(3) – any vector aligned with the direction of degeneracy, ***X***(6) – vector composed from the 11-, 12-, 13-, 22-, 23- and 33- elements of symmetric degenerate matrix of the quadratic form (37) to be reduced.

Output: ***T***(3,3) – generating matrix (38) of the transform required.

SUBROUTINE INVRST(e(3), th, T(3,3))

Input: ***e***(3) – any vector, ***th*** – angle in *degrees*.

Output: ***T***(3,3) – matrix transposed with respect to (38); can serve as a generating matrix of the transform restoring the laboratory coordinate representation.

SUBROUTINE YNR(x, n, Y(n))

Input: ***x*** – real argument, ***n*** – number of components in the output.

Output: vector containing $\boldsymbol{Y}(\boldsymbol{j})=\boldsymbol{e}^{-\boldsymbol{x}}\boldsymbol{I_{j-1}}(\boldsymbol{x})$, where $\boldsymbol{I_j}$ is the modified Bessel function of the first kind, $\boldsymbol{j}=1,\ldots,\boldsymbol{n}$.

SUBROUTINE INSIDE(x, y, z, in)

Input: ***x***, ***y***, ***z*** – coordinates of a point in *m* (for concordance with the `tj2` routine).

Output: ***in*** (logical): ***.true.*** if this point is inside the vessel, ***.false.*** otherwise.

SUBROUTINE SEGMENT(phi, R1, Z1, R2, Z2, e1(3), e2(3))

Input: ***phi*** – toroidal angle in *radians* of the section, (***R1***,***Z1***) and (***R2***,***Z2***) – two points (in *m*) in this section, which are presumably on different sides of the vessel surface.

Output: ***e1***(3), ***e2***(3) – coordinates (in *m*) of the pair of adjacent vertexes of the polygon representing the vessel section, such that these vertexes enclose the point of intersection with the line joining given points. If no intersection – two zero vectors.

SUBROUTINE CROSS(x1, y1, x2, y2, x3, y3, x4, y4, x, y, k)

Input: the line segment between the points (***x1***,***y1***) and (***x2***,***y2***), and the line segment between another two points (***x3***,***y3***) and (***x4***,***y4***), on a plane.

Output: (***x***,***y***) – coordinates of the intersection point, if any; ***k*** (integer) classifies the positional relationship of the given line segments: ***2*** – overlap, ***1*** – cross at a point, ***0*** – mutually disjoint and parallel, ***-1*** – mutually disjoint and non-parallel, ***-2*** – mutually disjoint though aligned.

SUBROUTINE CLspline(n, x, x1, x2, f1(n), f2(n), df1(n), df2(n), f(n))

Input: ***n*** – number of dimensions, ***x*** – coordinate of a point, between ***x1*** and ***x2***, ***f1***(***n***) and ***f2***(***n***) – values of a function at ***x1*** and ***x2***, ***df1***(***n***) and ***df2***(***n***) – values of a derivative at ***x1*** and ***x2***.

Output: ***f***(***n***) – the value of the spline interpolant $\boldsymbol{f} = \boldsymbol{F}_{\mathbf{C}} + (\boldsymbol{F}_{\mathbf{L}} - \boldsymbol{F}_{\mathbf{C}})^3 / \boldsymbol{F}_0^2$ at the given point, where $\boldsymbol{F}_{\mathbf{C}}$ is the cubic interpolant, $\boldsymbol{F}_{\mathbf{L}}$ is the linear interpolant, and $\boldsymbol{F}_0 = \max\{|\boldsymbol{f1}-\boldsymbol{f2}|, \boldsymbol{e}^{-1}\max\{|\boldsymbol{df1}|,|\boldsymbol{df2}|\}|\boldsymbol{x1}-\boldsymbol{x2}|\}$.

SUBROUTINE CSPL(n, x(n), y(n), Dy(2), C(3,n−1))

Input: ***n*** – number of tabulated values, ***x***(***n***) – array containing the data point abscissas (must be increasing), ***y***(***n***) – array containing the data point ordinates, ***Dy***(2) – array containing the values of ***dy***/***dx*** at the outmost points.

Output: ***C***(3,***n***−1) – array containing the cubic spline coefficients: $\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{y}(i) + \boldsymbol{C}(1,i)(\boldsymbol{x} - \boldsymbol{x}(i)) + \boldsymbol{C}(2,i)(\boldsymbol{x} - \boldsymbol{x}(i))^2 + \boldsymbol{C}(3,i)(\boldsymbol{x} - \boldsymbol{x}(i))^3$, $\boldsymbol{x}(i) \le \boldsymbol{x} < \boldsymbol{x}(i+1)$, $i = 1,\dots,\boldsymbol{n}-1$.

Calls to external libraries

TJ2LIB [11] or its substitute	**init_tj2_lib** **b_field_car** **flux_car** **grad_flux_car** **tj2** + dependencies
NAG [12]	**D02EJF** + dependencies
ODEPACK [13]	**LSODE** **LSODA** + dependencies

[11] V. Tribaldos, B. Ph. van Milligen, A. López-Fraguas. *TJ-II Library Manual.* Informes Técnicos CIEMAT **963** (2001).

[12] *NAG Fortran Library Manual.* http://www.nag.com/numeric/fl/manual/html/FLlibrarymanual.asp .

[13] A. C. Hindmarsh. *ODEPACK, A Systematized Collection of ODE Solvers*, in Scientific Computing, R. S. Stepleman et al. (eds.), North-Holland, Amsterdam (1983), 55–64.