

tr00ab

Calculates rotational transform given an arbitrary tangential field.

[called by: [dforce](#) and [mp00ac](#).]

contents

1 tr00ab	1
1.0.1 constructing straight field line angle on interfaces	1
1.0.2 alternative iterative method	1

1.0.1 constructing straight field line angle on interfaces

1. The algorithm stems from introducing a straight field line angle $\theta_s = \theta + \lambda(\theta, \zeta)$, where

$$\lambda = \sum_j \lambda_{o,j} \sin(m_j \theta - n_j \zeta) + \sum_j \lambda_{e,j} \cos(m_j \theta - n_j \zeta) \quad (1)$$

and insisting that

$$\frac{\mathbf{B} \cdot \nabla \theta_s}{\mathbf{B} \cdot \nabla \zeta} = \dot{\theta}(1 + \lambda_\theta) + \lambda_\zeta = \iota, \quad (2)$$

where ι is a constant that is to be determined.

2. Writing $\dot{\theta} = -\partial_s A_\zeta / \partial_s A_\theta$, we have

$$\partial_s A_\theta \iota + \partial_s A_\zeta \lambda_\theta - \partial_s A_\theta \lambda_\zeta = -\partial_s A_\zeta \quad (3)$$

3. Expanding this equation we obtain

$$\begin{aligned} & (A'_{\theta,e,k} \cos \alpha_k + A'_{\theta,o,k} \sin \alpha_k) \iota \\ & + (A'_{\zeta,e,k} \cos \alpha_k + A'_{\zeta,o,k} \sin \alpha_k) (+m_j \lambda_{o,j} \cos \alpha_j - m_j \lambda_{e,j} \sin \alpha_j) \\ & - (A'_{\theta,e,k} \cos \alpha_k + A'_{\theta,o,k} \sin \alpha_k) (-n_j \lambda_{o,j} \cos \alpha_j + n_j \lambda_{e,j} \sin \alpha_j) \\ = & - (A'_{\zeta,e,k} \cos \alpha_k + A'_{\zeta,o,k} \sin \alpha_k), \end{aligned} \quad (4)$$

where summation over $k = 1, \text{mn}$ and $j = 2, \text{mns}$ is implied

4. After applying double angle formulae,

$$\begin{aligned} & (A'_{\theta,e,k} \cos \alpha_k + A'_{\theta,o,k} \sin \alpha_k) \iota \\ & + \lambda_{o,j} (+m_j A'_{\zeta,e,k} + n_j A'_{\theta,e,k}) [+ \cos(\alpha_k + \alpha_j) + \cos(\alpha_k - \alpha_j)] / 2 \\ & + \lambda_{e,j} (-m_j A'_{\zeta,e,k} - n_j A'_{\theta,e,k}) [+ \sin(\alpha_k + \alpha_j) - \sin(\alpha_k - \alpha_j)] / 2 \\ & + \lambda_{o,j} (+m_j A'_{\zeta,o,k} + n_j A'_{\theta,o,k}) [+ \sin(\alpha_k + \alpha_j) + \sin(\alpha_k - \alpha_j)] / 2 \\ & + \lambda_{e,j} (-m_j A'_{\zeta,o,k} - n_j A'_{\theta,o,k}) [- \cos(\alpha_k + \alpha_j) + \cos(\alpha_k - \alpha_j)] / 2 \\ = & - (A'_{\zeta,e,k} \cos \alpha_k + A'_{\zeta,o,k} \sin \alpha_k), \end{aligned} \quad (5)$$

and equating coefficients, an equation of the form $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ is obtained, where

$$\mathbf{x} = \left(\underbrace{\iota}_{\mathbf{x}[1]}, \underbrace{\lambda_{o,2}, \lambda_{o,3}, \dots}_{\mathbf{x}[2:N]}, \underbrace{\lambda_{e,2}, \lambda_{e,3}, \dots}_{\mathbf{x}[N+1:2N-1]} \right)^T. \quad (6)$$

1.0.2 alternative iterative method

1. Consider the equation $\dot{\theta}(1 + \lambda_\theta) + \lambda_\zeta = \iota$, where $\lambda = \sum_j \lambda_j \sin \alpha_j$, given on a grid

$$\dot{\theta}_i + \dot{\theta}_i \sum_j m_j \cos \alpha_{i,j} \lambda_j - \sum_j n_j \cos \alpha_{i,j} \lambda_j = \iota, \quad (7)$$

where i labels the grid point.

2. This is a matrix equation . . .