Box 7-2. *(continued)*

The colatitude θ_E is stored instead of the latitude λ_E because it is θ_E that is actually used (Box 7-1).
The second subroutine (ROTATE) performs the rotation:

	Stack				Registers		
Function	x	y	z	t	R_{11}	R_{12}	R_{13}
LBL "ROTATE"	r	ϕ	λ	-	θ_E	ϕ_E	Ω
RCL 12	ϕ_E	r	ϕ	λ	θ_E	ϕ_E	Ω
CHS	$-\phi_E$	r	ϕ	λ	θ_E	ϕ_E	Ω
ST + Z	$-\phi_E$	r	ϕ_1	λ	θ_E	ϕ_E	Ω
CLX	0	r	ϕ_1	λ	θ_E	ϕ_E	Ω
RCL 11	θ_E	r	ϕ_1	λ	θ_E	ϕ_E	Ω
CHS	$-\theta_E$	r	ϕ_1	λ	θ_E	ϕ_E	Ω
XEQ "ROT 2"	r	ϕ_2	λ_1	$-\theta_E$	θ_E	ϕ_E	Ω
RCL 13	Ω	r	ϕ_2	λ_1	θ_E	ϕ_E	Ω
ST+ Z	Ω	r	ϕ_3	λ_1	θ_E	ϕ_E	Ω
CLX	0	r	ϕ_3	λ_1	θ_E	ϕ_E	Ω
RCL 11	θ_E	r	ϕ_3	λ_1	θ_E	ϕ_E	Ω
XEQ "ROT 2"	r	ϕ_4	λ'	θ_E	θ_E	ϕ_E	Ω
RCL 12	ϕ_E	r	ϕ_4	λ'	θ_E	ϕ_E	Ω
ST+ Z	ϕ_E	r	ϕ'	λ'	θ_E	ϕ_E	Ω
CLX	0	r	ϕ'	λ'	θ_E	ϕ_E	Ω
RCL 13	Ω	r	ϕ'	λ'	θ_E	ϕ_E	Ω
R↓	r	ϕ'	λ'	Ω	θ_E	ϕ_E	Ω
XEQ "S-C"	x'	y'	z'	Ω	θ_E	ϕ_E	Ω
XEQ "C-S"	r	ϕ'	λ'	Ω	θ_E	ϕ_E	Ω
RETURN	r	ϕ'	λ'	Ω	θ_E	ϕ_E	Ω

The "S-C" and "C-S" conversions between spherical and Cartesian coordinates might seem to negate each other, but they perform the important function of converting ϕ' to the range $-180° \leq \phi' \leq 180°$. The operation ST+ Z, CLX was used instead of the equivalent operation "ROT 3" because it takes fewer bytes of memory and operates faster.

Box 7-3. How to Rotate Using a Computer.

If you would rather work with algebra than geometry, you can do finite rotations using a 3×3 matrix. If point **A** is a vector with global Cartesian coordinates (A_x, A_y, A_z) prior to rotation, then the components (A_x', A_y', A_z') of **A** after rotation to **A′** may be found from the matrix multiplication

$$\mathbf{A}' = \mathbf{R}\,\mathbf{A}$$

where **R** represents a 3×3 matrix. Writing all of the terms of the vector and matrix we have

(continued)