```
% Steger Warming Flux
% Primitive state variable vector V_i
% Conservative far-field state variable vector W_∞
% outward normal n_i∞

function F = steger_warming_flux(V_i, n_iinf)

W_i = V_to_W(V_i);

gam = 1.4;
l = sqrt(n_iinf(1)^2 + n_iinf(2)^2);

% construct unit normal vector
tilde = n_iinf / l;
tilde_nx = tilde(1);
tilde_ny = tilde(2);

rho = V_i(1);
u = V_i(2);
v = V_i(3);
p = V_i(4);

% normal velocity
vn = u * n_iinf(1) + v * n_iinf(2);
c = sqrt(gam * p / rho);

Dp = [max(0, vn); max(0, vn); max(0, vn + c * l); max(0, vn - c * l)];
Dm = [min(0, vn); min(0, vn); min(0, vn + c * l); min(0, vn - c * l)];

theta = tilde_nx * u + tilde_ny * v;
phi = sqrt((gam - 1) / 2 * (u * u + v * v));
beta = 1 / (2 * c * c);

Q = [[1                 0                             1↙
1];
    [u                 tilde_ny                      (u + tilde_nx * c)↙
(u - tilde_nx * c)];
    [v                 -tilde_nx                     (v + tilde_ny * c)↙
(v - tilde_ny * c)];
    [phi * phi / (gam - 1)   tilde_ny * u - tilde_nx * v   (phi * phi + c * c) / (gam -↙
1) + c * theta   (phi * phi + c * c) / (gam - 1) - c * theta]];

Qinv = [[1 - phi * phi / (c * c)             (gam - 1) * u / c^2                   (gam↙
- 1) * v / c^2                   -(gam - 1) / c^2];
    [-(tilde_ny * u - tilde_nx * v)   tilde_ny                            -tilde_nx↙
0];
    [beta * (phi^2 - c * theta)               beta * (tilde_nx * c - (gam - 1) * u)↙
beta * (tilde_ny * c - (gam - 1) * v)    beta * (gam - 1)];
    [beta * (phi^2 + c * theta)               -beta * (tilde_nx * c + (gam - 1) * u)   -↙
beta * (tilde_ny * c + (gam - 1) * v)   beta * (gam - 1)]];


fp = (0.5 * rho / gam) * [(2.0 * (gam - 1) * Dp(1) + Dp(3) + Dp(4));
    (2.0 * (gam - 1) * Dp(1) * u + Dp(3) * (u + c * tilde_nx) + Dp(4) * (u - c *↙
tilde_nx));
    (2.0 * (gam - 1) * Dp(1) * v + Dp(3) * (v + c * tilde_ny) + Dp(4) * (v - c *↙
tilde_ny));
    ((gam - 1) * Dp(1) * (u * u + v * v) + 0.5 * Dp(3) * ((u + c * tilde_nx)^2 + (v + c *↙
```

```
tilde_ny)^2) + 0.5 * Dp(4) * ((u - c * tilde_nx)^2 + (v - c * tilde_ny)^2) + (3.0 - gam)↙
* (Dp(3) + Dp(4)) * c * c / (2 * (gam - 1)))];

fm = Q * (Dm .* (Qinv * W_i));

F = fp + fm;
end
```