Table of Contents


```
function exitVelocity = nozzle(pStagIn, tStagIn, pExit, efficiency)
```

nozzle

Finds the exhaust velocity of a compressible gas exiting a nozzle.

Syntax

```
%#
% exitVelocity = nozzle(pStagIn, tStagIn, pExit, efficiency);
%
```

Desctiption

This function uses basic relationships of compressible flow through a nozzle to solve for the exit velocity. It assumes that the flow is isentropic and behaving as an ideal gas

- pStagIn - stagnation pressure of the gas at nozzle input [Pa]
- tStagIn - stagnation temperature of the gas at nozzle input [K]
- efficiency - float representing efficiency of nozzle (0 < eta <=1)
- pExit - pressure at exit of nozzle [Pa]
- exitVelocity - velocity of fluid at nozzle exit [m/s]

Example

```
%#
% P0 = 0.697e6; % [Pa]
% T0 = 372;     % [K]
% pAtm = 101.3; % [Pa]
% eta = 0.95;
% exhaustVelocity = nozzle(P0, T0, pAtm, eta)
%
```

See Also

compressor, cobustor, turbine, fan

Code

```
exitVelocity = NaN;

% Constants
gamma = 1.4;
R = 287; % [J/kg/K]

% To see if the nozzle is choked compare exit pressure to critical p
pCritical = 0.528 * pStagIn;
if (pCritical < pExit) % choked
    tCritical = tStag / (1 + (gamma-1)/2);
    tExit = tCritical;
    exitVelocity = sqrt(gamma * R * tExit);
else    % not choked
    constant = (pStagIn/pExit)^((gamma-1)/gamma) - 1;
    MaExit = sqrt(2/(gamma-1) * constant);
    tExit = tStagIn / (1+ (gamma-1)/2*MaExit^2);
    exitVelocity = MaExit*sqrt(gamma*R*tExit);
end

% scale by efficiency factor
exitVelocity = exitVelocity * efficiency;

Not enough input arguments.

Error in nozzle (line 39)
pCritical = 0.528 * pStagIn;

end
```

Published with MATLAB® R2017b