ThermoElectric Power System Simulator (TEPSS)

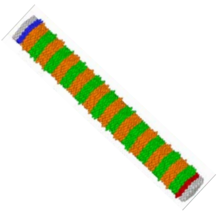
Sponsored by NYSERDA's Industrial Research
and Development Program

Robert J. Stevens
Mechanical Engineering
Rochester Institute of Technology

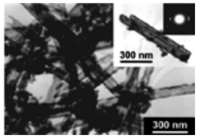
2011 DOE Thermoelectrics Applications Workshop
January 3 - 6, 2011
San Diego, CA

RIT Sustainable Energy Lab

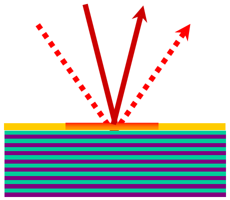
Overall Goal: To model, test, and design thermoelectric modules and systems for current and future power generation applications.



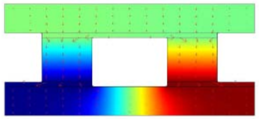
Nanoscale Heat Transfer



Material Science & Engineering (nano vs. bulk)



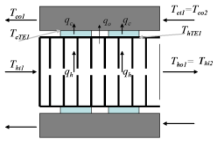
Material Testing



Device Modeling

Module Testing

Module Design



System Modeling

System Design & Testing

Fundamental ⟷ Applied

Overview

- Motivation
- TEPSS Overview/Architecture
- Component and Node Modeling
- Optimization Shell
- Thermoelectric Power Unit Component
- Proof of Concept
- Conclusion/Remarks

Motivation

Benefits

- Modular, scalable (mW → kW…)
- Solid-state, no moving parts
- Operate over a range of temperatures
- Transient thermal sources
- Minimal maintenance
- No noise, vibration

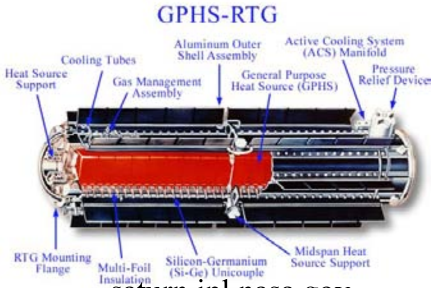
Niche Applications

- Car exhaust
- Woodstoves, cookstoves
- Remote power
- Sensors
- Radioisotope thermoelectric generator

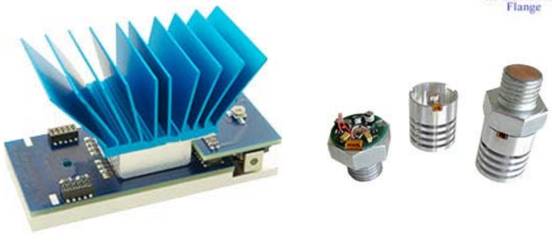
greencar.com

globalte.com

TEGPower.com



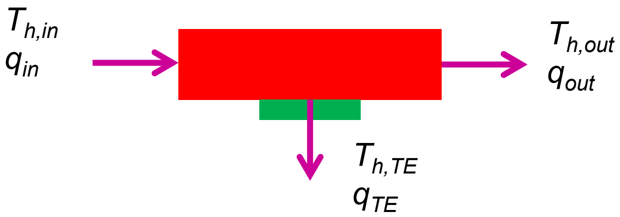
saturn.jpl.nasa.gov



micropelt.com

Motivation

- Over 50 quadrillion BTUs of waste heat generated each year.
- Recovering just 1% would power New York indefinitely.
- Improved TE materials, increased efficiency →smaller heat sinks.
- Trade-offs:
  - high ZT vs. cost
  - high ΔT vs. ΔP, weight, volume, cost



- Currently specific device and system models exist.
- Need to integrate material properties, engineering thermal modeling, and economics.

TEPSS Project Goal

Create a versatile tool to evaluate whether or not thermoelectrics are currently or will soon become technically and economically viable for a specific application and if so determine what the optimal system might look like.

- The tool should help quickly assess a range of potential applications for waste heat recovery using emerging thermoelectric materials.
- Most current thermoelectric modeling is geared towards very specific applications and may not consider system trade-offs.
- Historically module and system design have been loosely coupled.

TEPSS Overview

Requirements

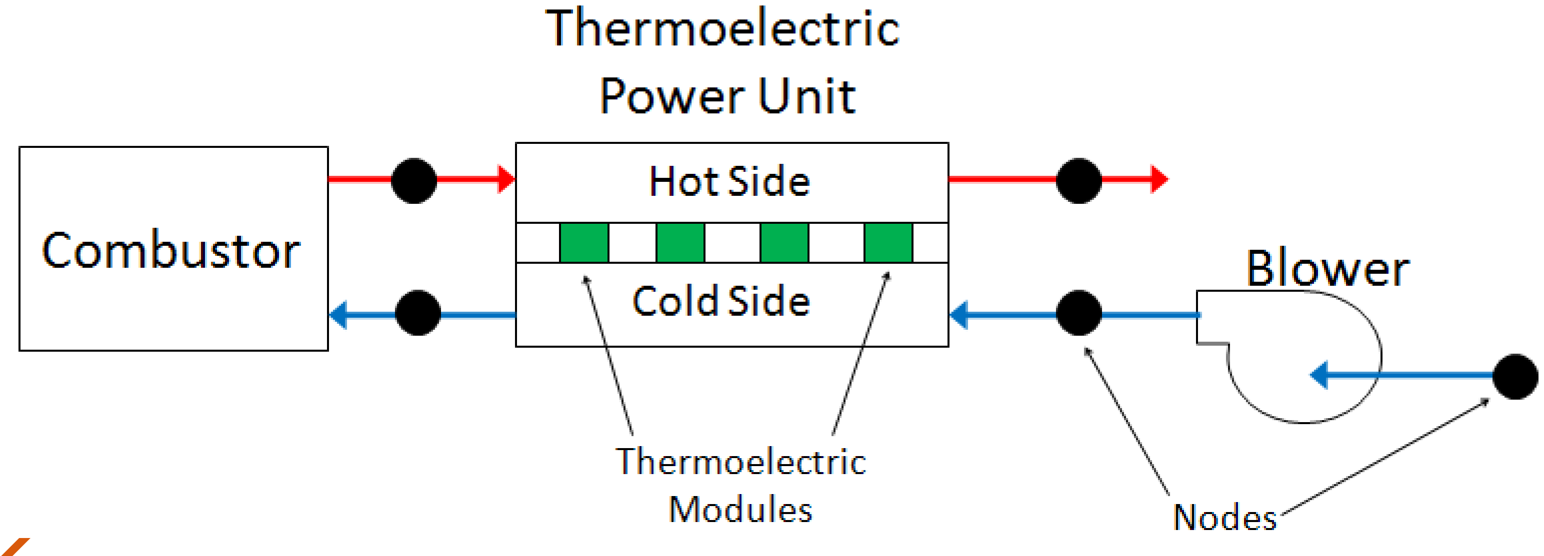
- Solve system of equations for the system steady operating state
- Unlimited system concepts defined by user
- Objective function is defined by the user
- Optimizes system configuration with respect to user defined design variables
- Open source and expandable
- Easy to use, modify, and reuse

Challenges

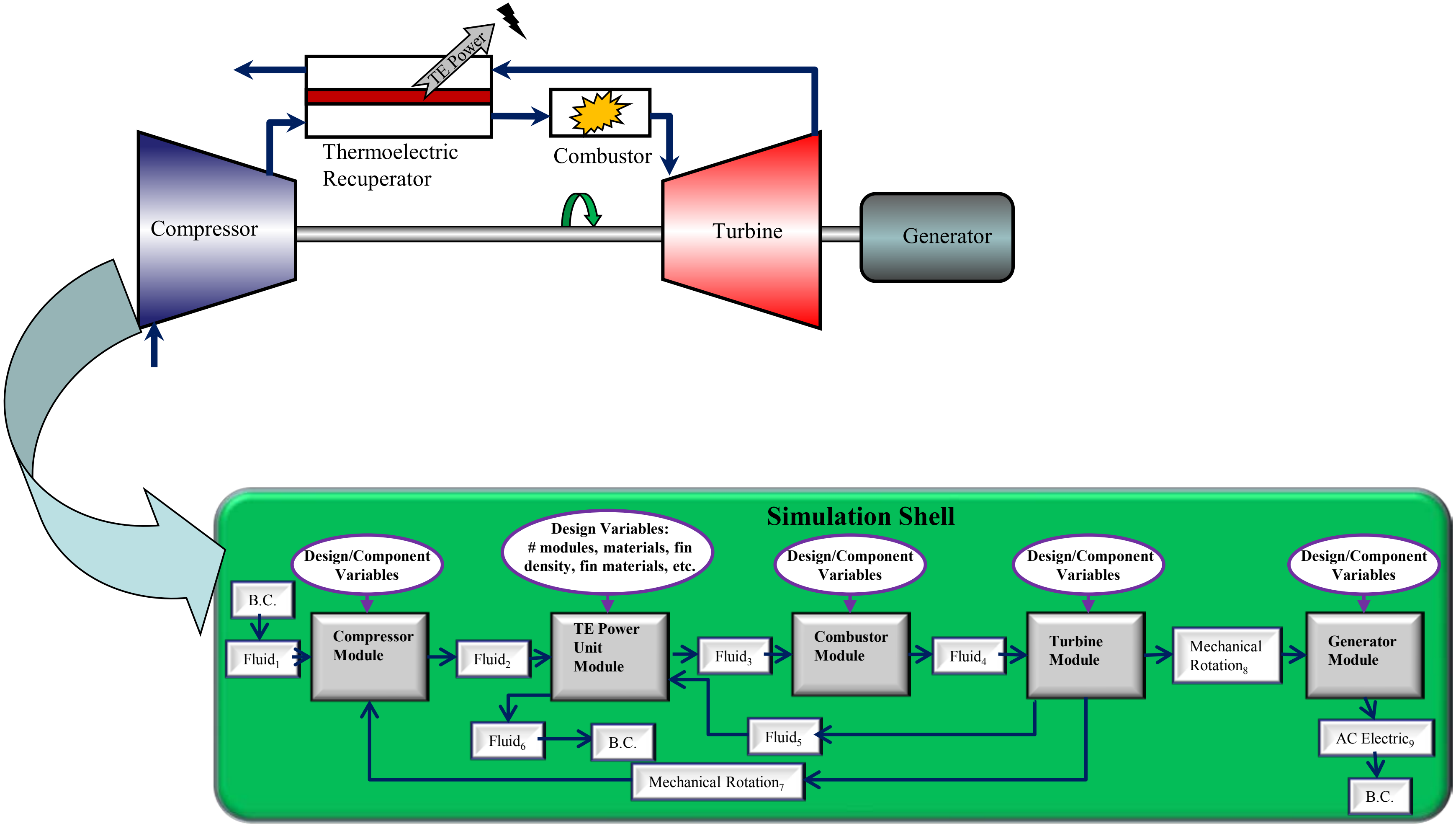
- Energy components are a combination of empirical, analytical or FE/FD models
- Often highly nonlinear system of component models
- System of equations changes for each user defined concept

TEPSS Architecture

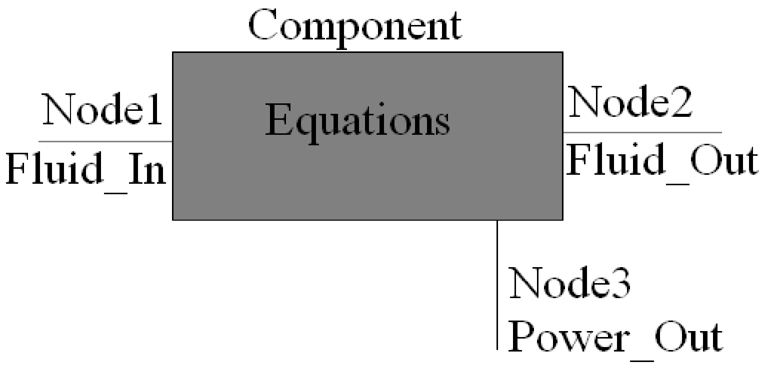
- What are energy systems?
  - Independent (modular) components
    - Components contain models and independent of rest of system
  - Interconnected by nodes
    - Nodes store data



TEPSS Architecture



Component



- Object oriented programming (Matlab)
- Components contain engineering models
  - Mass, energy conservation, performance data
- Components are linked together by nodes
  - Nodes provide component boundary conditions.
  - Nodes belong to a specific *domain* (fluid, mechanical rotation, electrical, etc.).

Component Equations and Errors

Conservation of mass

$$\dot{m}_{in} - \dot{m}_{out} = 0$$

$$\dot{m}_{in} - \dot{m}_{out} = e(1)$$

Conservation of energy

$$Q_{in} - \dot{m}Cp\Delta T = 0$$

$$Q_{in} - \dot{m}Cp\Delta T = e(2)$$

Viscous dissipation

$$\Delta P + k\dot{m}^2 = 0$$

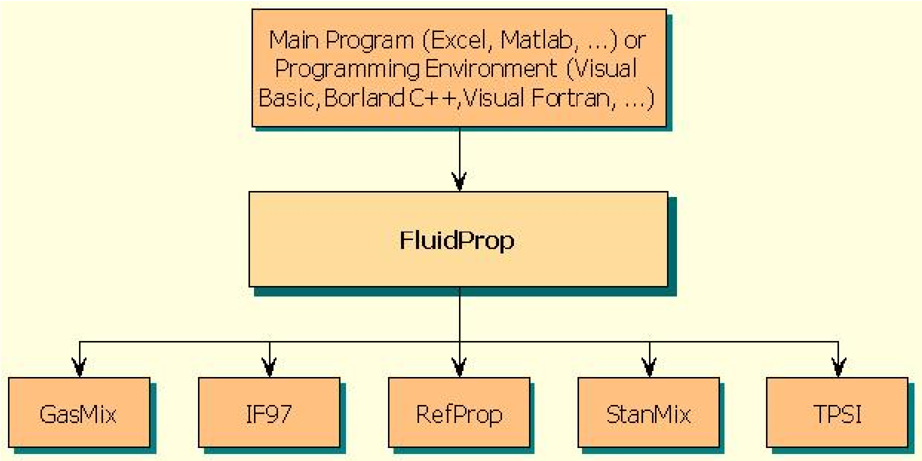
$$\Delta P + k\dot{m}^2 = e(3)$$

Collect errors, use root finding algorithm to set = 0.

$$e = [e(1), e(2), e(3), \ldots, e(n)]$$

Node Domains

- Through and Across variables
- Fluid domain (fluidprop.tudelft.nl/)



Domain	Variable
electrical	current
	voltage
fluid	mass flow
	specific enthalpy
	pressure
rotation mechanical	torque
	angular velocity

Steady State Simulation

Systems of nonlinear equations solved iteratively with Newton's Method:

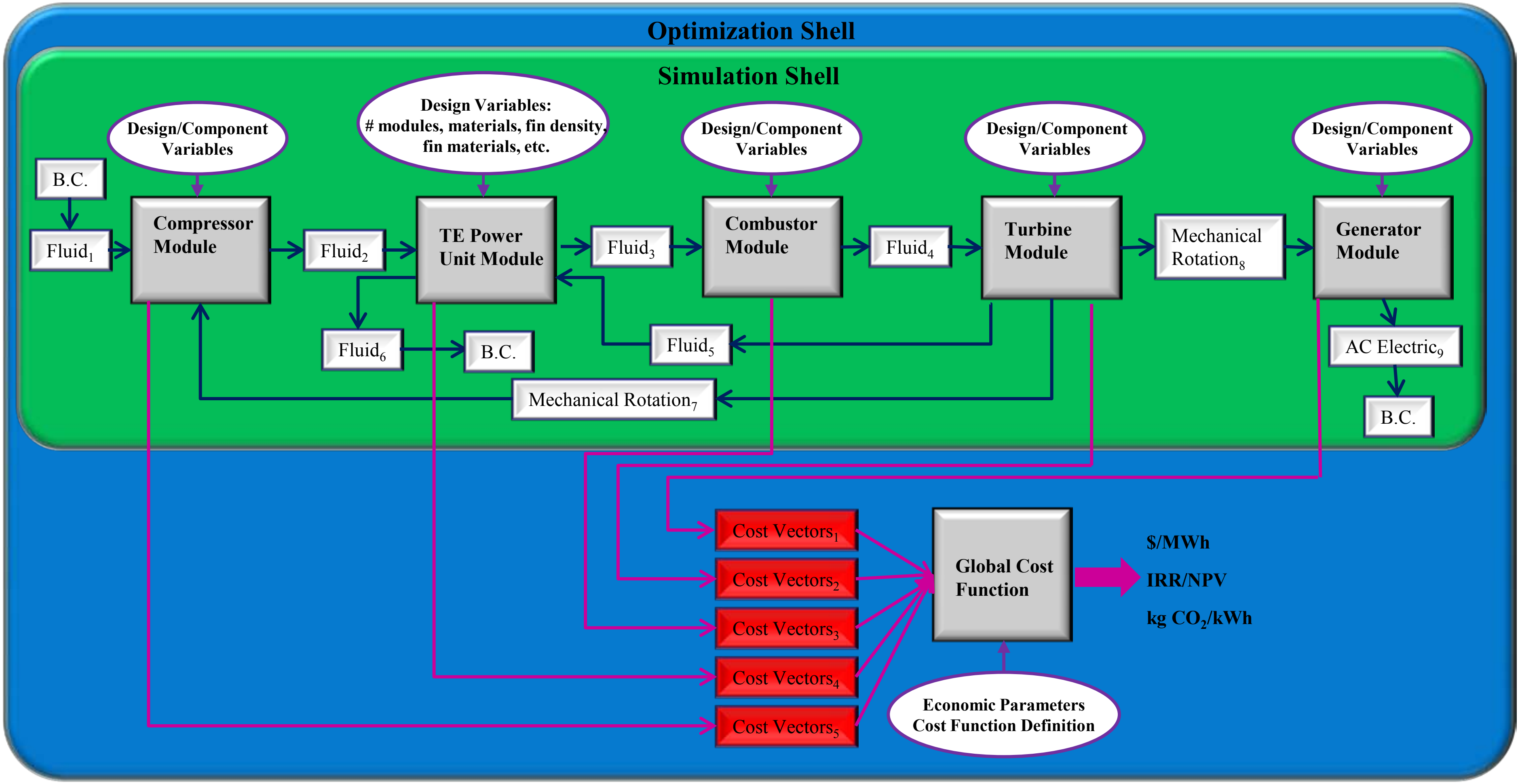
$$\Delta x_i = -[J(x_i)]^{-1} \times f(x_i)$$

$$x_{i+1} = x_i + \Delta x_i$$

$$[J(x_i)] = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_m} \end{bmatrix}$$

i is iteration number, x and f are vectors, $[J(x_i)]$ is Jacobian Matrix at x_i

TEPSS Architecture



Optimization Shell

Simulation Shell

Design/Component Variables

Design Variables: # modules, materials, fin density, fin materials, etc.

Design/Component Variables

Design/Component Variables

Design/Component Variables

B.C.

Fluid$_1$

Compressor Module

Fluid$_2$

TE Power Unit Module

Fluid$_3$

Combustor Module

Fluid$_4$

Turbine Module

Mechanical Rotation$_8$

Generator Module

Fluid$_6$

B.C.

Fluid$_5$

Mechanical Rotation$_7$

AC Electric$_9$

B.C.

Cost Vectors$_1$

Cost Vectors$_2$

Cost Vectors$_3$

Cost Vectors$_4$

Cost Vectors$_5$

Global Cost Function

Economic Parameters Cost Function Definition

\$/MWh

IRR/NPV

kg CO_2/kWh

General Cost Function

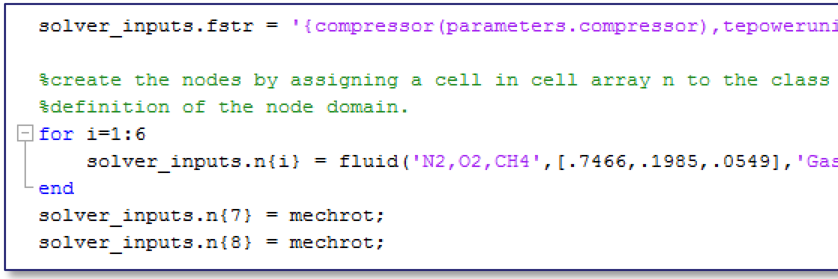
- Common cost metrics
  - Simple payback (added cost/net savings per year)
  - Cost per unit energy output
  - Emissions per unit energy output

$$\text{cost} = \frac{\sum_{i=1}^{n} A_i + t\sum_{i=1}^{n} B_i + t\sum_{i=1}^{n}([C_i][U])}{\sum_{i=1}^{n} D_i + t\sum_{i=1}^{n} E_i + t\sum_{i=1}^{n}([F_i][V])} + \sum_{i=1}^{n} \Phi$$

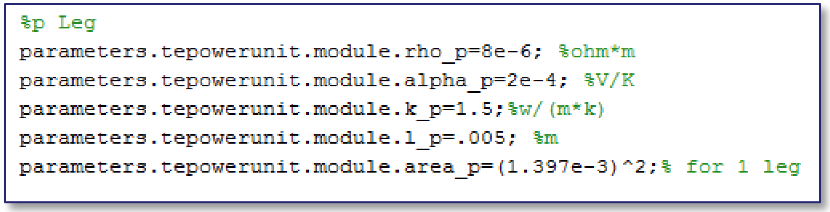
n = number of components
A-F, U, V and t = user inputs
Φ = user defined penalty function

User Inputs

- Solver inputs
  - components and nodes
  - interconnections (system concept)
  - initial guesses
  - boundary conditions
  - convergence criteria
- Component parameters
- Optimization inputs
  - design variables
  - upper and lower bounds
  - convergence criteria
- Cost function

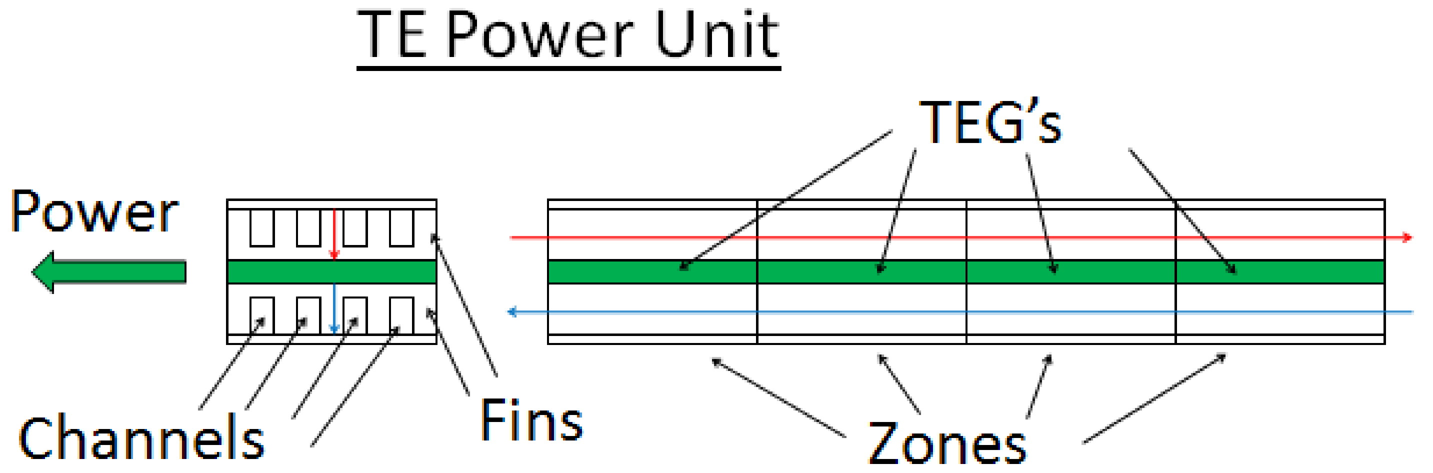


```
solver_inputs.fstr = '{compressor(parameters.compressor),tepoweruni

%create the nodes by assigning a cell in cell array n to the class
%definition of the node domain.
for i=1:6
    solver_inputs.n{i} = fluid('N2,O2,CH4',[.7466,.1985,.0549],'Gas
end
solver_inputs.n{7} = mechrot;
solver_inputs.n{8} = mechrot;
```

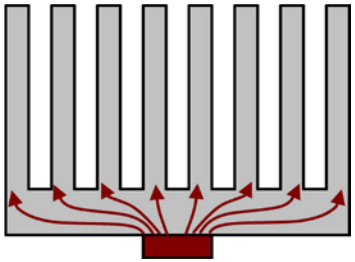


```
%p Leg
parameters.tepowerunit.module.rho_p=8e-6; %ohm*m
parameters.tepowerunit.module.alpha_p=2e-4; %V/K
parameters.tepowerunit.module.k_p=1.5;%w/(m*k)
parameters.tepowerunit.module.l_p=.005; %m
parameters.tepowerunit.module.area_p=(1.397e-3)^2;% for 1 leg
```

Thermoelectric Power Unit Component



- Counter and parallel flow using any combination of fluids as well as isothermal or constant heat flux configurations.
- Independently specified heat sink type and geometry (aligned and staggered plates, pin fins), material, fin pitch, and contact resistances.
- Pressure drop models for all configurations
- Thermal coupling with surrounding environment.
- Module layout in zones, series, and parallel.
- Heat spreading accounted for using modified model developed by Ellison. (G.N.Ellison, *IEEE Trans. on Comp. & Packaging Tech.*, 2003)



Thermoelectric Power Unit Zone

- 12 equations for each zone represent various energy balances and the standard TEM models
- Set of equations are nonlinear and solved internally using numerical techniques to obtain heat flows and temperatures throughout the system and final power recovered by each zone

$$q_{ins(j)} = \frac{T_{H(j)} - T_{C(j)}}{R_{ins}}$$

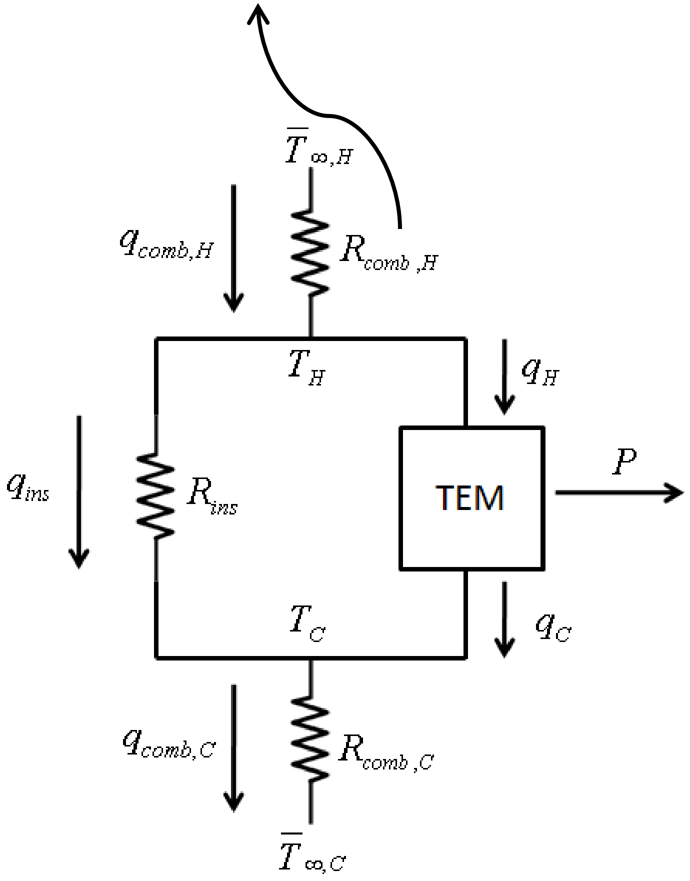
$$q_{comb,H(j)} = \frac{\Delta T_{lm,H(j)}}{R_{cond,H}}$$

$$q_{H(j)} = \frac{T_{H(j)} - T_{C(j)}}{R_{th}} - \frac{1}{2} I_{(j)}^2 R_e + \alpha T_{H(j)} I_{(j)}$$

$$q_{C(j)} = \frac{T_{H(j)} - T_{C(j)}}{R_{th}} + \frac{1}{2} I_{(j)}^2 R_e + \alpha T_{H(j)} I_{(j)}$$

$$I_{(j)} = \frac{\alpha\left(T_{H(j)} - T_{C(j)}\right)}{R_e + R_{load}}$$

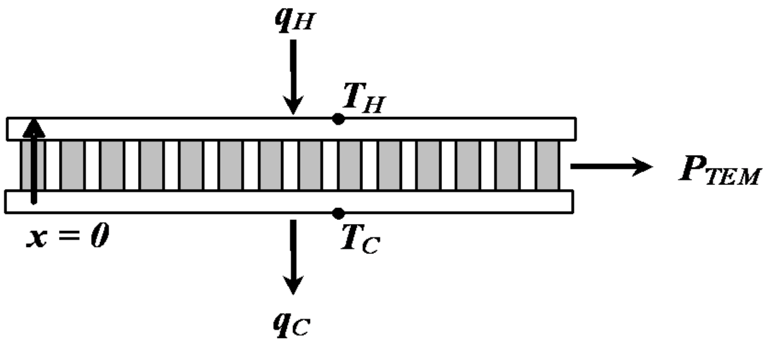
R_{comb} = f(fin type & material, geometry, flow rate, fluid type, heat spreading)



Standard 1D Thermoelectric Model

- Multiple module model options:

$$q_H = I\,\alpha_{p,n} T_H + K(T_H - T_C) - \frac{I^2 R_e}{2}$$



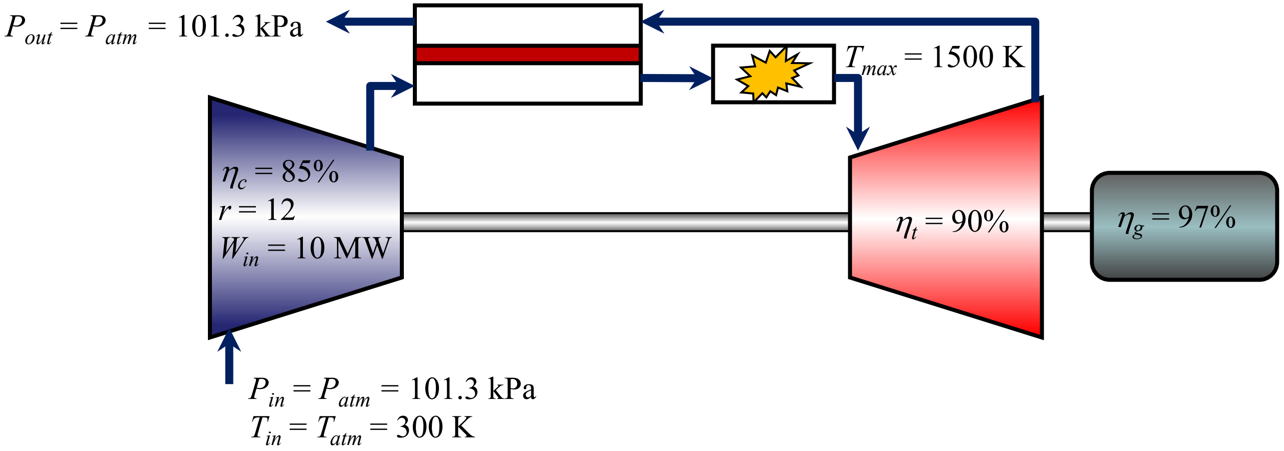
  - experimentally measured values
  - manufacturer specifications
  - build from selected TE material properties, leg geometries, and contact resistances.

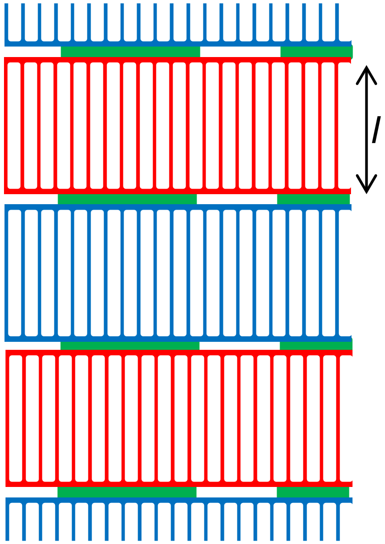
$$K = n\left(\frac{A_p}{2R''_c + {l_p}/{\lambda_p}} + \frac{A_n}{2R''_c + {l_n}/{\lambda_n}}\right)$$

$$R_e = n\left(\frac{\rho_p l_p + 2r''}{A_p} + \frac{\rho_n l_n + 2r''}{A_n}\right)$$

$$\alpha_{p,n} = n(\overline{\alpha_p} - \overline{\alpha_n})$$

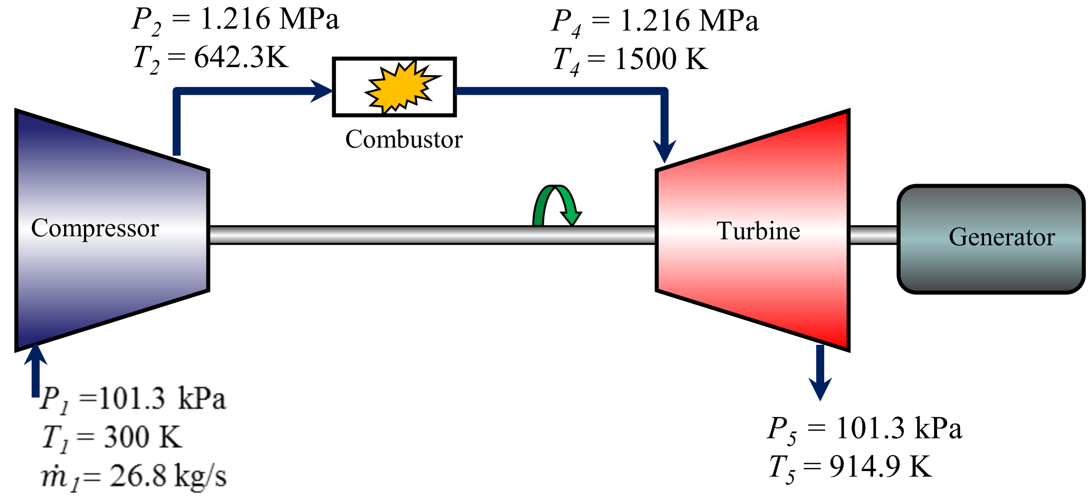
Proof of Concept





- Power Unit Design Parameters:
  - 1.0 mm aligned plate fins, counter flow configuration
  - 10 zones, 12,500 - 4cm x 4cm modules (127 couples)
  - TE Properties: $\alpha_p = -\alpha_p$ = 200 μV/K, $\lambda_p = \lambda_p$ = 1.5 W/m·K, $\rho_p = \rho_p$ = 8e-6 Ω ·m
    ZT ~ 2.5 @ 750K
  - Base area to module ratio of 9
  - Design parameters: channel height (3-12 cm) and fin density (4-24/module)
- Costs (materials, per leg pair, per module, fuel cost, 20 year @ 100% capacity)
- Cost function = additional cost per generated kWh$_e$ above base system

Proof of Concept



$P_2 = 1.216$ MPa
$T_2 = 642.3$K

$P_4 = 1.216$ MPa
$T_4 = 1500$ K

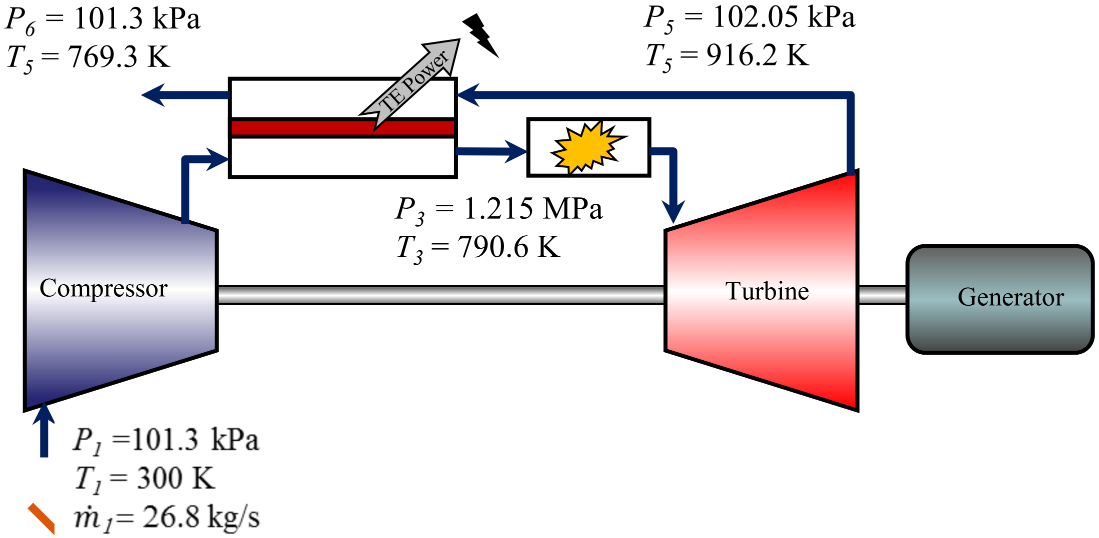
$P_1 = 101.3$ kPa
$T_1 = 300$ K
$\dot{m}_1 = 26.8$ kg/s

$P_5 = 101.3$ kPa
$T_5 = 914.9$ K

Base Case:

$W_{net} = 10.00$ MW

$Cost = 4.69$ ¢/kWh



$P_6 = 101.3$ kPa
$T_5 = 769.3$ K

$P_5 = 102.05$ kPa
$T_5 = 916.2$ K

$P_3 = 1.215$ MPa
$T_3 = 790.6$ K

$P_1 = 101.3$ kPa
$T_1 = 300$ K
$\dot{m}_1 = 26.8$ kg/s

Optimal Case:

***l* = *6.03* cm**

***n* = *13.9* fins/module**

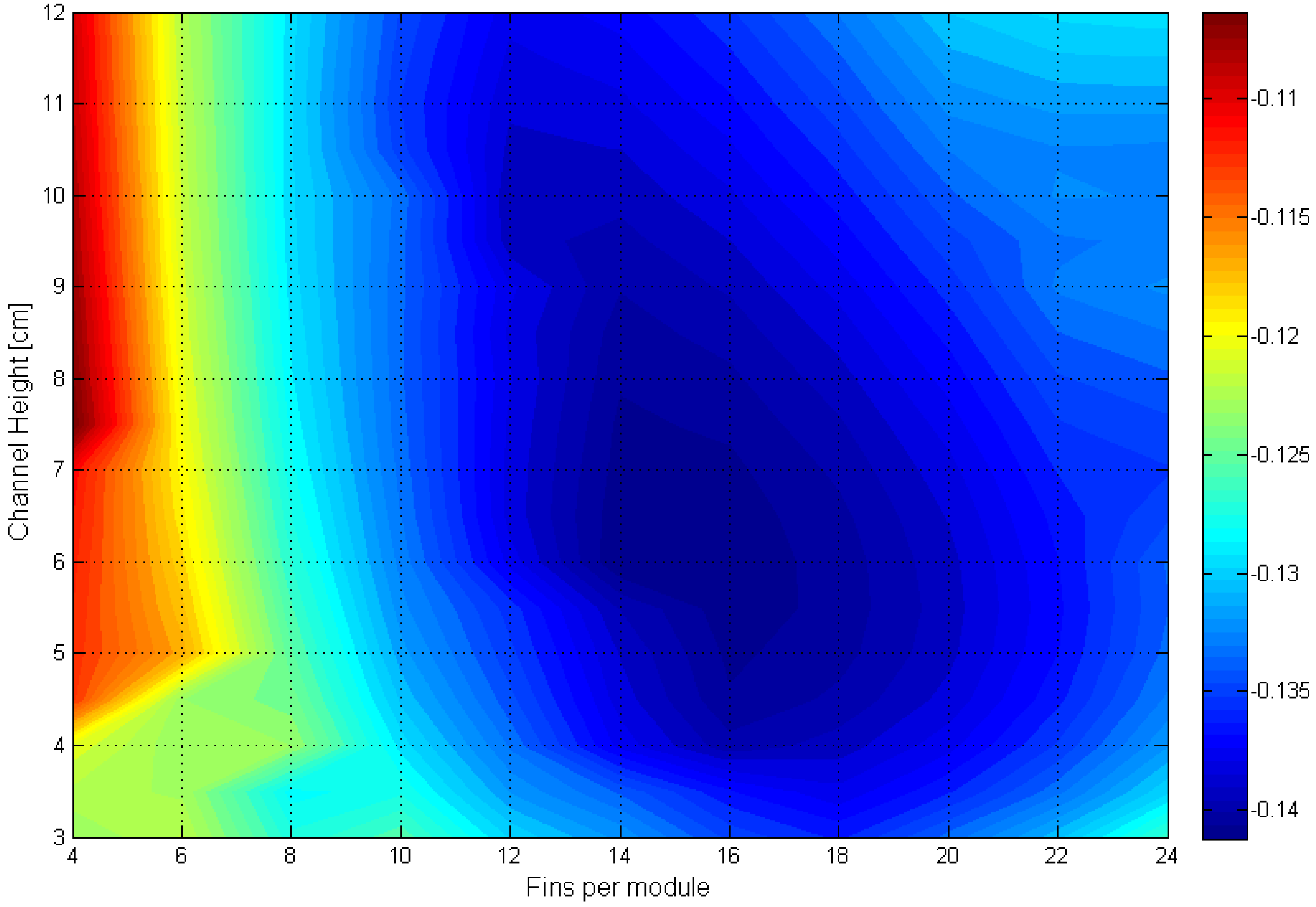
$W_{net} = 10.11$ MW

$W_{TEG} = 121$ kW

$Cost = 4.03$ ¢/kWh

Cost function = -14.12%

Proof of Concept



Closing Remarks

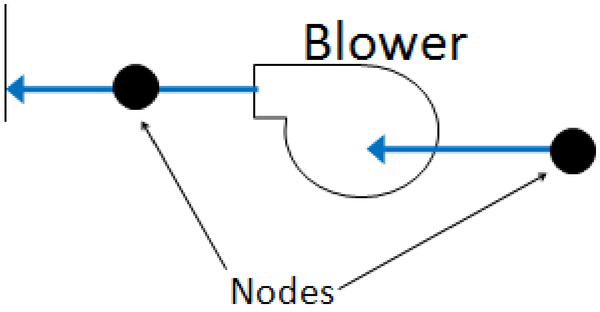
- TEPSS environment has been developed for simulating and optimizing thermoelectric power generation systems.
- TEPSS is expandable with reusable and customizable components.
- New node domains and components can be added to increase potential system concepts.
- TEPSS will allow for the exploration of suitable applications for emerging TE materials while coupling module level design with system level performance and economics.
- Working on more robust steady state solver and optimization options and developing more advanced component models.
- Interest in using/testing contact rjseme@rit.edu

Acknowledgements

- John Kreuder
- Andy Freedman
- NYSERDA (Contract # 11135)

Sample Component Class

```
classdef pump < handle

  properties (SetAccess=private)
    design variables
    component operating parameters
  end

  methods
    function obj = pump(parameters)
      component setup
    end

    function e = compute(obj, node1, node2, onoff)
      engineering models
    end

    function component_cost = cost(obj)
      component costing models
    end

    function y = paramcheck(obj)
      constraint and physical check models
    end
  end
end
```



Optimization in TEPSS

- MATLAB *fmincon* optimization algorithm
  Constrained nonlinear optimization
    - User supplies cost function
    - Simulation determines steady state
    - Cost function is evaluated at steady state
    - New parameters are chosen by *fmincon*
    - Repeat until cost function is minimized