Finite Element Analysis of Solid Oxide Fuel Cells: Coupled Electrochemistry, Thermal and Flow Analysis in *MARC*®

Zijing Lin, Moe Khaleel

Pacific Northwest National Laboratory

Richland, WA

Wayne Surdoval, Don Collins

National Energy Technology Laboratory

Pittsburgh, PA

August 29, 2003

COURSE OUTLINE

- Introduction
- Modeling tool requirements
- Technical approach
- The electrochemistry module
- SOFC modeling in the *MARC* code
- Sample Results
- EC module CPU time
- Ongoing and future work
- Summary

Introduction:

SOFC needs to compete with other heat engines

1. Higher Efficiency - optimization of plate design, stack configurations and operating conditions.
2. Understanding of the Complex Interactions between various electrochemical phenomena.
3. Improved Durability - Electrochemical current and chemical process generates heat, and thermal stresses affects structural reliability.
4. Lowered Manufacturing Cost - Modeling and simulation can be used to assist in finding structurally sound designs that are easy to manufacture and cost effective materials that meet operation requirements.

SECA core modeling goal

Develop versatile, robust modeling tools for industrial teams to speed up the SOFC technology development

Modeling tool requirements

Versatile multi-physics analysis methods

- The thermal, chemical, electrochemical and transport processes are strongly coupled.

Computational efficiency is critical

- Design optimization requires parametric studies of various geometric, material and operation parameters.
- The cost of such parametric studies increases exponentially with the number of the working parameters. Many parameters exist.

Accessible to the industrial teams.

- To maximizing the benefit of the SECA program, the modeling tool needs to be accessible to the industrial teams.

Technical Approach

Commercial finite element code as underlying platform

- Commercial codes provide the shortest path to well developed, multi-function tools that are widely accessible.
- Finite element analysis has a long history of structural design.
- FEA with Implicit algorithm is numerically stable and efficient.

***MARC* Capabilities**

- Multi-physics, robust structural (mechanical, thermal, flow) analysis tool.
- User-defined functions allow efficient algorithms describing the chemical, electrochemical processes and fluid flow to be incorporated.
- Electrochemistry, chemical reaction, heat generation and flow solution are based upon an in-house developed software package.

Electrochemistry Module

General description:

- Electrochemistry based on continuum level I-V relations: two models (Chick model and Virkar model) are implemented for flexibility and choice.
- Chemical reaction (water-gas shift, CH_4 internal reforming) based on equilibrium theory.
- Flow solution based upon assumption of laminar flow, taking conservation law into consideration.
- Distributed heat flux calculated according to respective mechanisms.

Continuum Level Electrochemistry Modeling

- Model for calculating current density, cell voltage, and heat production in a (SOFC) stack with H_2 or other fuels, taking as inputs local values of the gas partial pressures and temperatures. This approach is based on existing I-V relations.
- Virkar model – current generated by H_2 oxidation:

$$V(i) = E_{open} - iR_i - b\,\sinh^{-1}(i/2i_0) + (RT/4F)\ln(1-i/i_{O2}) + (RT/2F)\ln(1-i/i_{H2}) - (RT/2F)\ln[1+p^0_{H2}i/(p^0_{H2O}i_{H2})]$$

- Experiment data for different operating temperatures are used to determine R_i, b, i_0, $i_{H2}(=2FD_{eff}(T)p^0_{H2}/(RTl_a))$

I-V Model II for Composite Fuel (the Chick model)

$$V(i) = E_{open} - i\,R_i - b\,\sinh^{-1}(i/2i_0) + C\ln(1-i/i_{O2}) + 2C\ln(1-i_1/i_{H2}) - 2C\ln[1+i_1/i_{H2O})]$$

$$= E_{open} - iR_i - b'\sinh^{-1}(i/2i_0) + C\ln(1-i/i_{O2}) + 2C\ln(1-i_2/i_{CO}) - 2C\ln[1+i_2/i_{CO2})]$$

$$-\eta_{ac} = 2C\ln(1-i_1/i_{H2}) - 2C\ln(1+i_1/i_{H2O}) = 2C\ln(1-i_2/i_{CO}) - 2C\ln(1+i_2/i_{CO2})$$

$$t = \exp(-\eta_{ac}/2C) = [-B+(B2-4AC)^{1/2}]/(2A)$$

$$A = i/(i_{H2O}\ i_{CO2}) + 1/i_{H2O} + 1/i_{CO2}$$

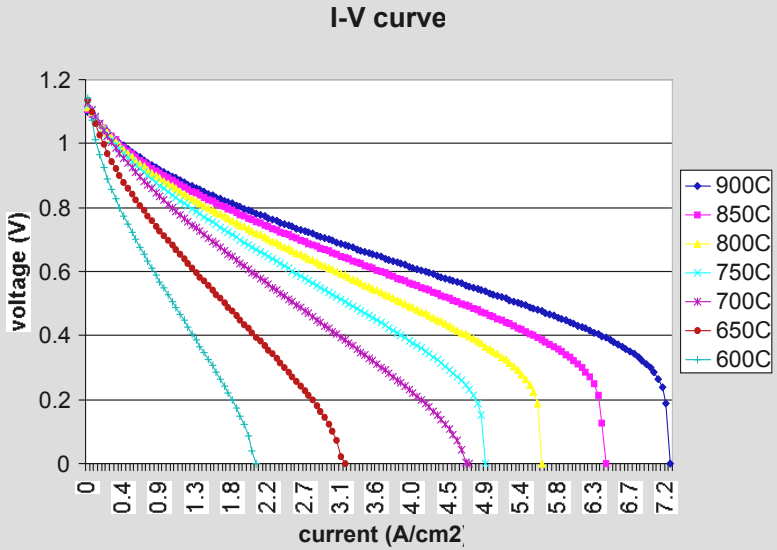
$$B = i/(i_{H2O}\ i_{CO}) + I/(i_{H2}\ i_{CO2}) + 1/i_{H2} + 1/i_{CO} - 1/i_{H2O} - 1/i_{CO2};\quad C = i/(i_{H2}\ i_{CO}) - 1/i_{H2} - 1/i_{CO})$$

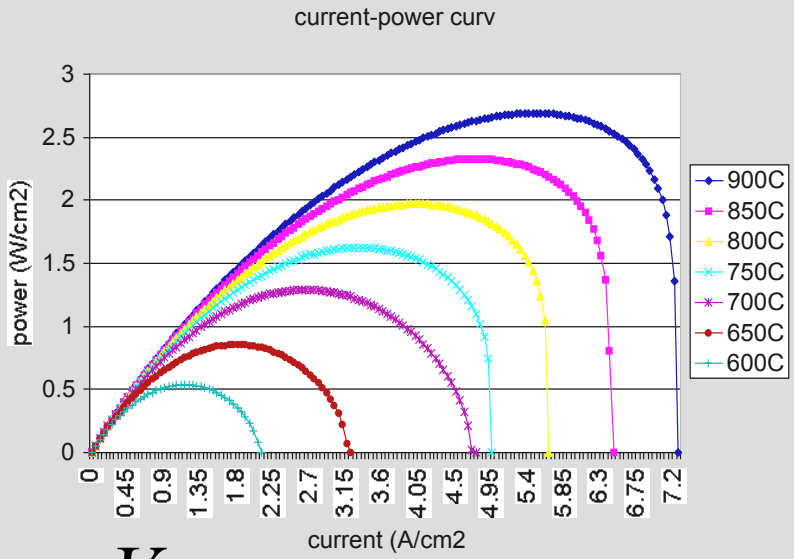
$$i_1 = (1-t)/(1/i_{H2} + t/i_{H2O});\quad i_2 = (1-t)/(1/i_{CO} + t/i_{CO2})$$

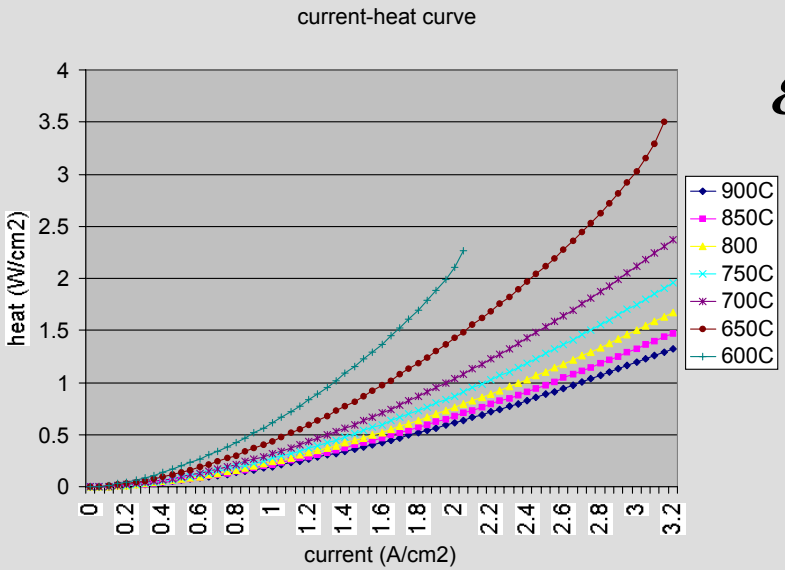
a. Limiting current determined by molecular diffusion theory for the gas mixture.

b. Flow content determined by equilibrium theory.

I-V curves: illustrative examples







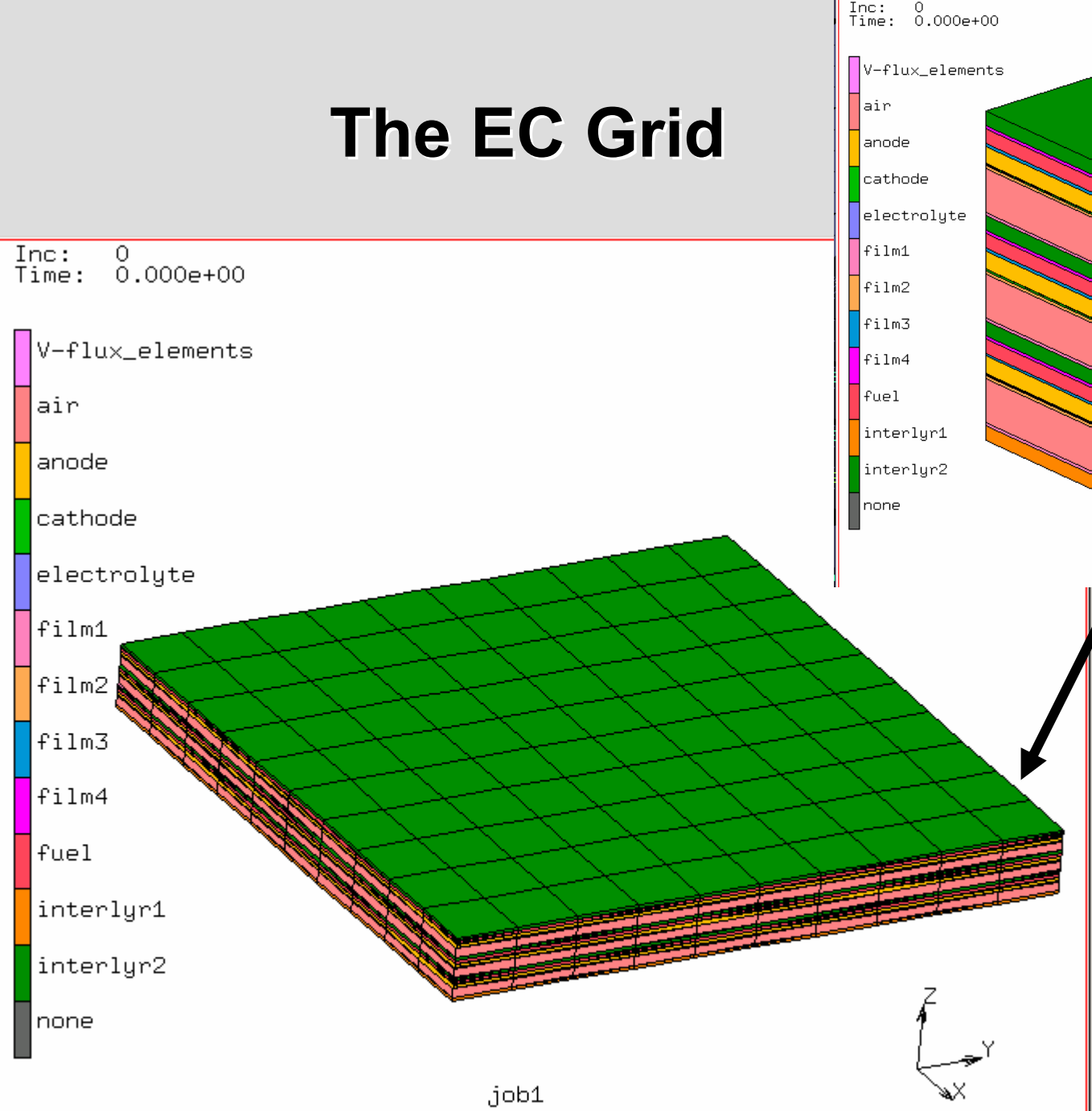
$$\varepsilon = \frac{\Delta G}{\Delta H}\frac{V_{op}}{V_{th}}U$$

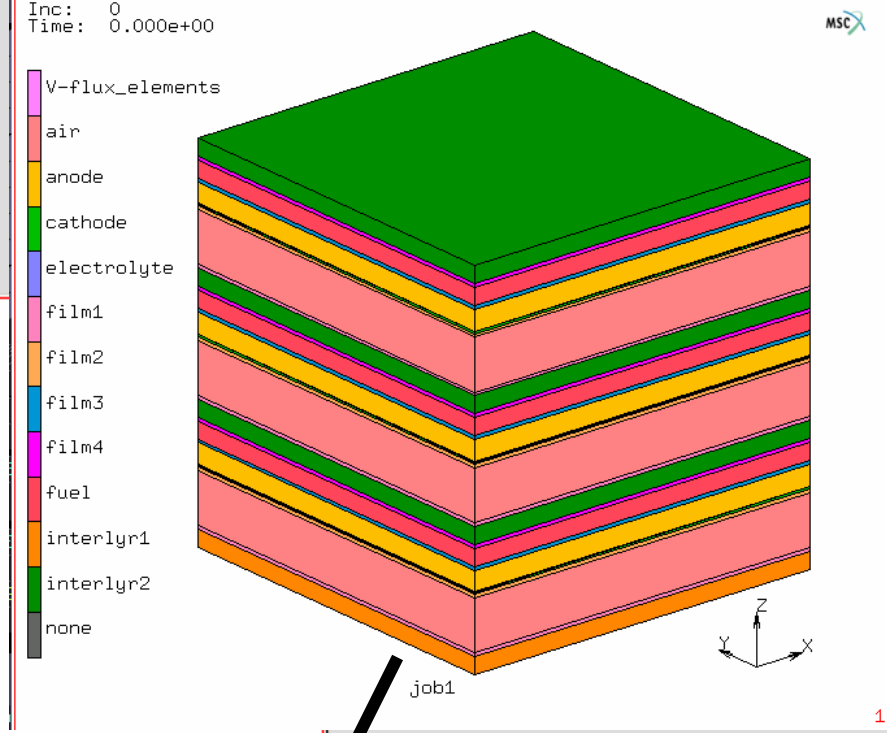
ε: fuel efficiency
ΔG: change of free energy
ΔH: change of enthalpy
V_{op}: operating voltage
V_{th}: Nernst potential
U: fuel utilization

The Electrochemistry Computational Grid

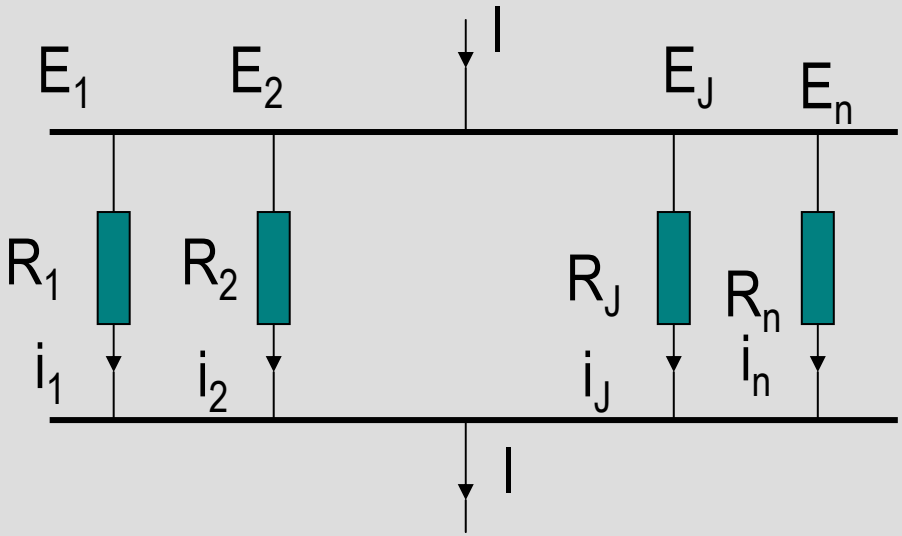
- Grid cells are defined in x and y in the plane of the active cell area.
- For a given x, y location, the current is constant over the z-grid.
- The I-V equation is used to compute cell voltage if current density is given, or to find the current density consistent with a given cell voltage.
- All cells in the stack have the same grid for the EC active area, but corresponding grid points in different cells can have different current densities.
- Distributed heat generation is determined over the grid cells for the various terms in the I-V equation, according to their respective physical origins.

The EC Grid


Inc: 0
Time: 0.000e+00
V-flux_elements
air
anode
cathode
electrolyte
film1
film2
film3
film4
fuel
interlyr1
interlyr2
none
Z
Y
X
job1
1


Inc: 0
Time: 0.000e+00
MSC
V-flux_elements
air
anode
cathode
electrolyte
film1
film2
film3
film4
fuel
interlyr1
interlyr2
none
Z
Y
X
job1
1

Single Cell Operation



"Effective" Joul Heat generated

$$Q = \sum_j Q_j = \sum_j i_j^2 R_j = \sum_j i_j (E_{o_j} - V)$$

Heat due to thermal irreversibility:

$$Q^{TS} = \sum_j Q_j^{TS} = \sum_j T_j \Delta S_j$$

Local open voltage E_j^o is a function of T, gas composition, etc.

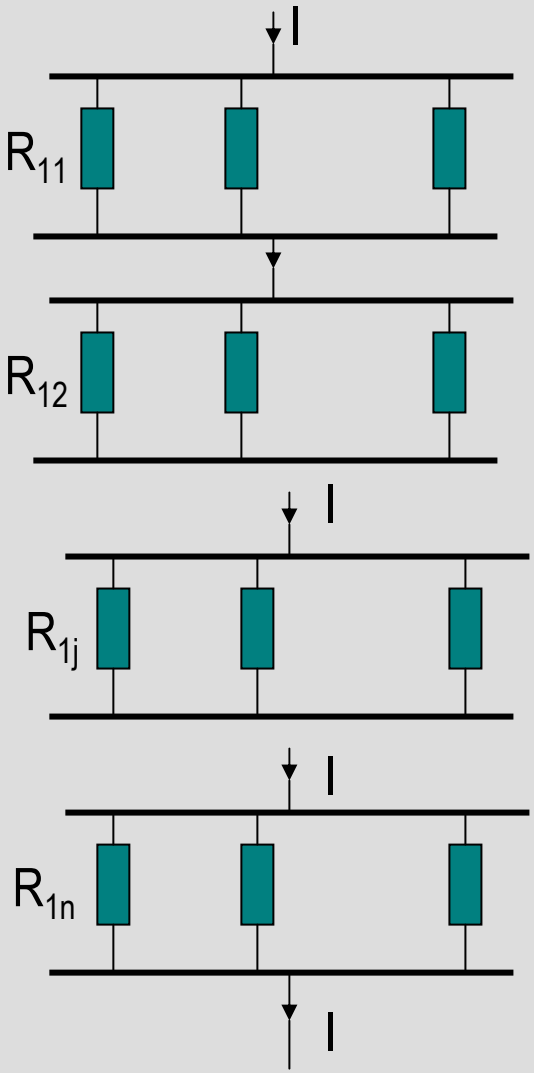
Current through the cell (the integration of current through all elements) I= $\sum_{j=1}^{n} i_j$

Working (output) voltage has only one value

$$V = E_{o_1} - i_1 R_1 = E_{o_2} - i_2 R_2 = \cdots = E_{o_n} - R_n i_n$$

$$that \quad is \quad i_j = \left(\frac{E_{o_j} - V}{R_j} \right)$$

Stack Operations



- Total current through any cell in the stack is the same and equals the output current

- Output stack voltage is the sum of the cell working voltages

$$V_{stack} = \sum_{j=1}^{m} V_j$$

- "Effective" Joul Heat generated

$$Q = \sum_j Q_j = \sum_j \sum_k i_{kj} \left(E^o_{kj} - V_j\right)$$

- Heat due to thermal irreversibility is added to get the total heat production

Functionality of the I-V user relation

- Cell current is specified - The cell voltage is determined according to the equi-potential condition of the electrodes. The resulting current density distribution is integrated to the specified total current.
- Total stack current is specified - Individual cell potential is determined separately as in the previous point, and their sum gives the total output voltage.
- Cell voltage is specified - Current density distribution is determined by I-V equation with bisection search.
- Total stack voltage is specified - All the cell voltages are adjusted such that their sum is the specified value while the total current is the same for each cell.

The above is done self-consistently and the resulting current distribution, heat productions, and cell voltages are all returned to the user.

Flow solution: Mass balance with boundary condition

$dn_k/dt = r_k dv - u dc_k$; u: volume flow rate; c_k: species mole fraction; r_k: reaction rate

Steady state: $dn_k/dt = 0$, for H_2 gives $c_{H_2} = c^0_{H_2} - \int_0^l \frac{Ia}{2uF} dx$

I: current density, a: channel width for the grid of interest,

l: channel length

Other gas components can be obtained in a similar way

Fuel utilization $U=I_{tot}/2uFC_{f0}$, specifying U is equivalent to specifying I_{tot} => 3rd code option for I-V specification

Self Consistence

I (V) on grids $\leftrightarrow$ local gas partial pressures

- The initial guess current density by average T, V_{cell} and inlet gas pressure and flow rate.
- The current density distribution determines the gas pressure changes in the channels.
- The resulting gas pressures determines the new current density.
- Mixing new and old current distribution to get revised guess of the current distribution.
- Check convergence.

EC module structure

EC module interface subroutine

↓

Initial guess of current density and pressure distribution

↓

Compute current densities, cell voltages, species concentrations and iterate to self-consistency

↓

Compute various other quantities for output

EC Module Summary

- The E.C. module is capable of computing current density distribution, chemical and electrical heat generation and fuel cell efficiency.

- It can address the dependence of fuel cell performance on the operating Temperature, fuel composition, gas pressures, flow rate and SOFC geometries and configurations.

- It has been linked to *MARC*, and it can be linked to other simulation software to improve modeling quality e.g., provide FEA with E.C. heat to improve thermal stress analysis.

SOFC Modeling with *MARC*

- The E.C. module is a single subroutine called by *MARC*:

  udf_qiv (ivflag, itot, vtot, ufuel, nD, ivpower, heattot, iflowtype, nfueltype, nspecies, lxcell, lycell, lzcell, aair, bair, lair, dair, afuel, bfuel, lfuel, dfuel, ncell, vcell, ngridx, ngridy, ngridair, fuelvxn, airvn, pin, pout, dn_gas, tairxyn, tfuelxyn, tairn, tfuelxn, ngridz, ngridzi, txyzn, pxyn, ixyn0, ixyn, qxyzn, ch2xyn, po2xyn_an, uo2, gridx, gridy, gridz, dzgrid, qohmic_tot, qts_tot, qchem_tot, q_tot, ivchange)

- The EC module is interfaced with *MARC* by a user-defined subroutine, flux() ("udf_qiv" is called by "flux").
- The SOFC geometry, material property, boundary condition, operation parameters (fuel/air flow, working temperature, etc) and mesh are generated in *Mentat* (the *MARC* pre- and post-processor).
- The Flow pattern can be cross-, co-, or counter-flow.
- The Heat transfer mechanisms considered are convection and conduction.
- The SOFC model can be a single cell or a multi-cell stack.
- Modeling both steady state and startup (transient).

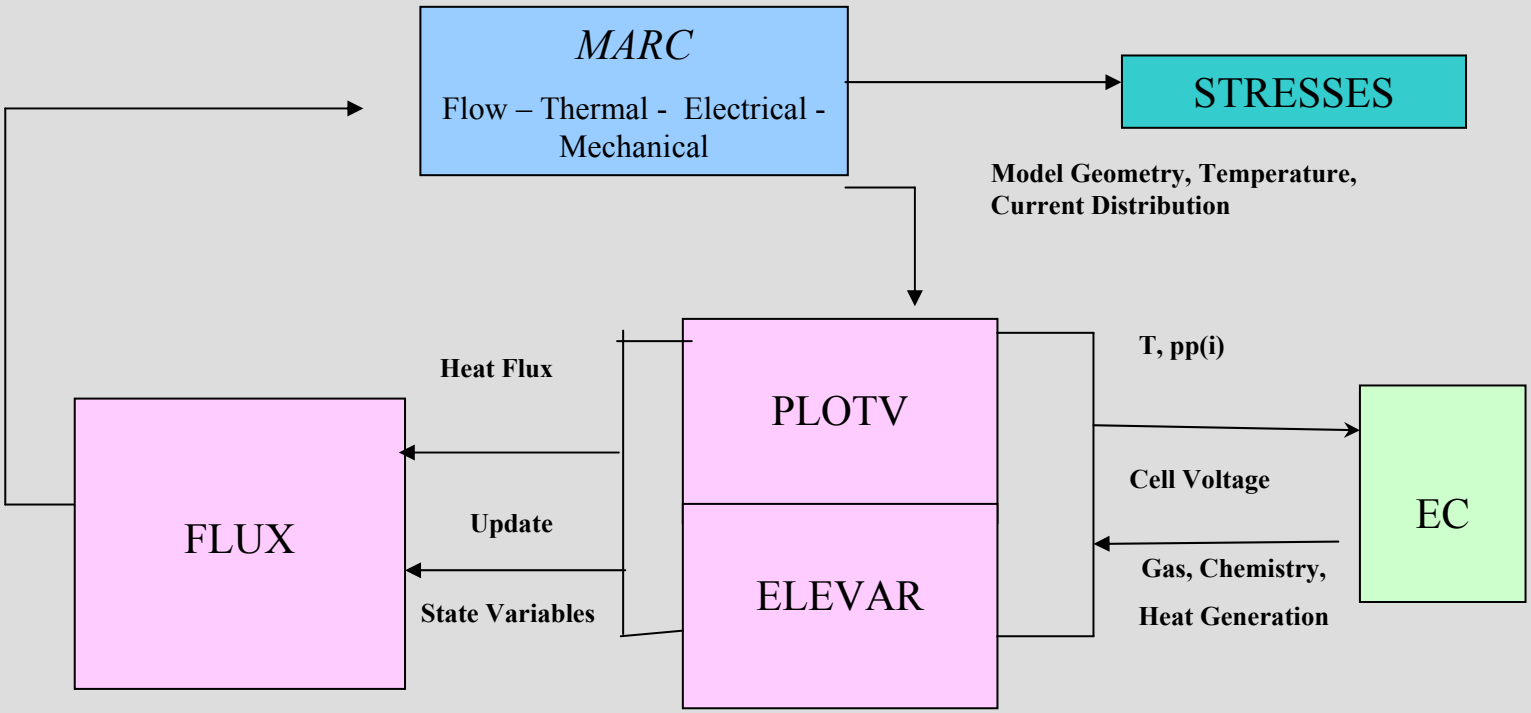
Linkage between EC module and *MARC*

MARC source subroutine (flux(f, ts, n, time))

Input temperature profile and geometry info ↓ flux and other

↑ return heat state variables

EC module interface subroutine



Running the *MARC*-EC tool

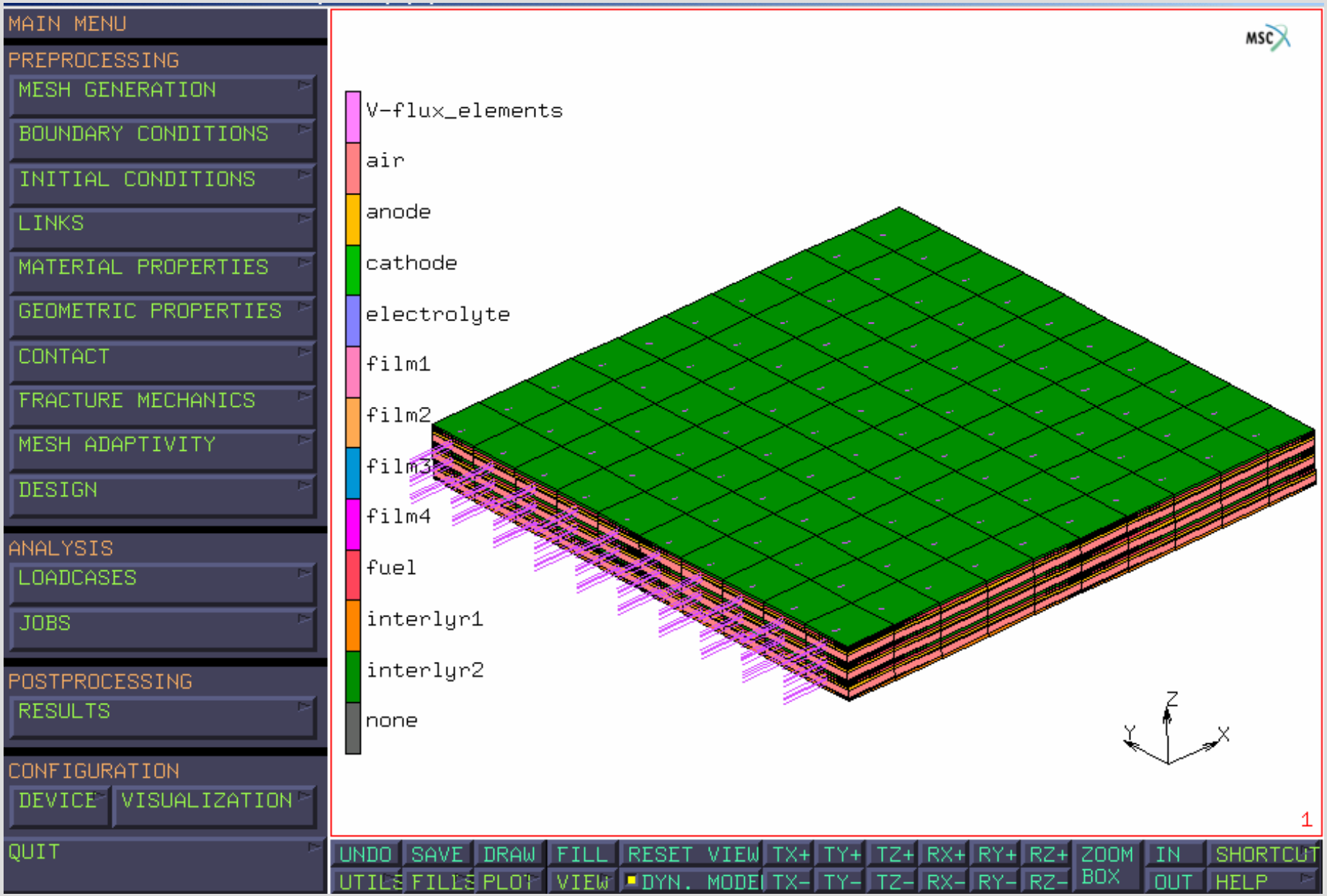
1. Generate the stack model in a *Mentat* procedure file. Include the file name for the user subroutine in the procedure file.
2. The procedure file includes the material properties, boundary conditions, operation parameters, etc.
3. Start Mentat (as *MARC* graphic user interface tool) program:

   Mentat2003 &
4. Execute the procedure file in *Mentat*:

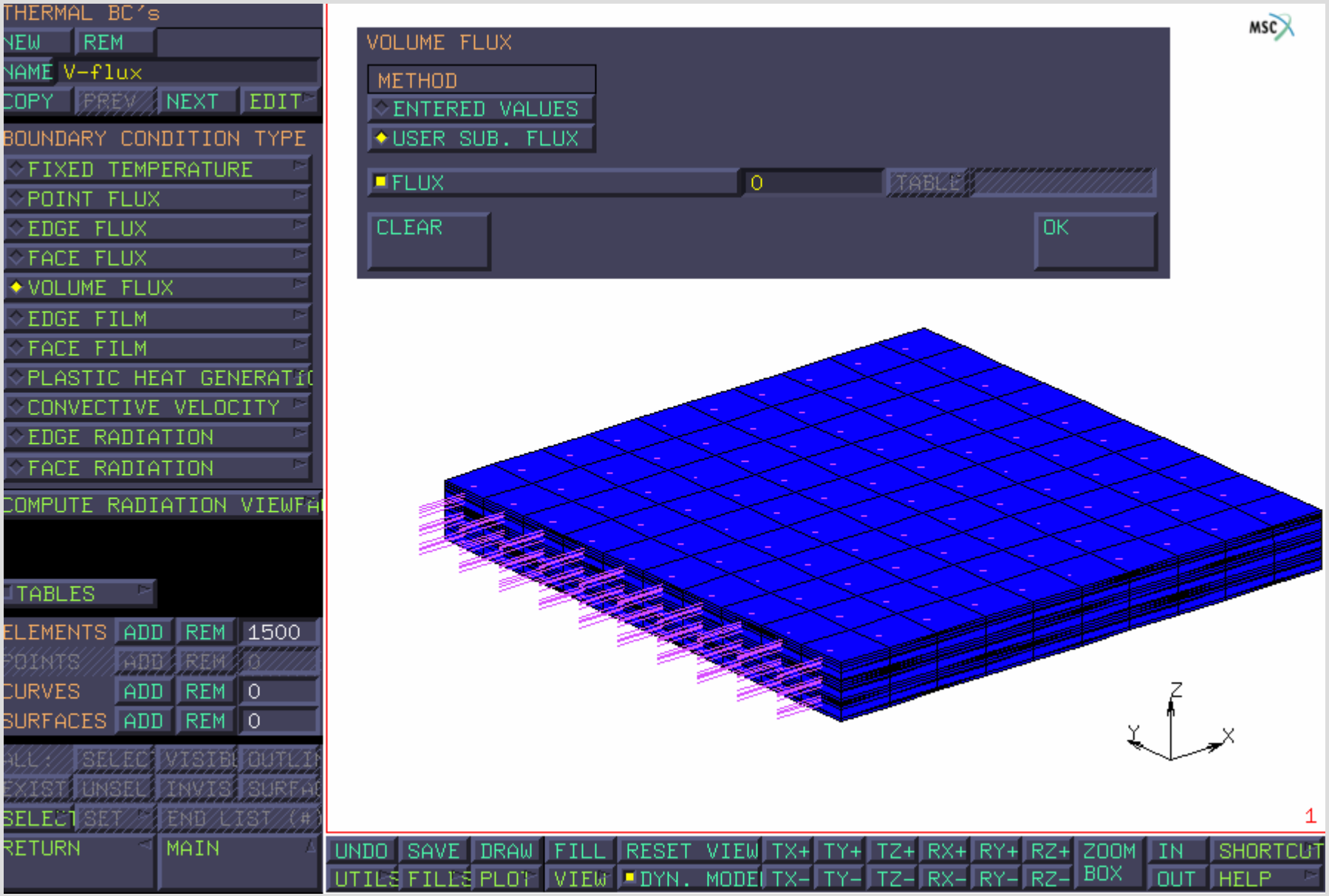
   Main menu/Util/Procedures/Execute file_name/OK.

   “Fill” to view the model.
5. Summit job: Main menu/JOBS/RUN/Summit.
6. Analyze the results.

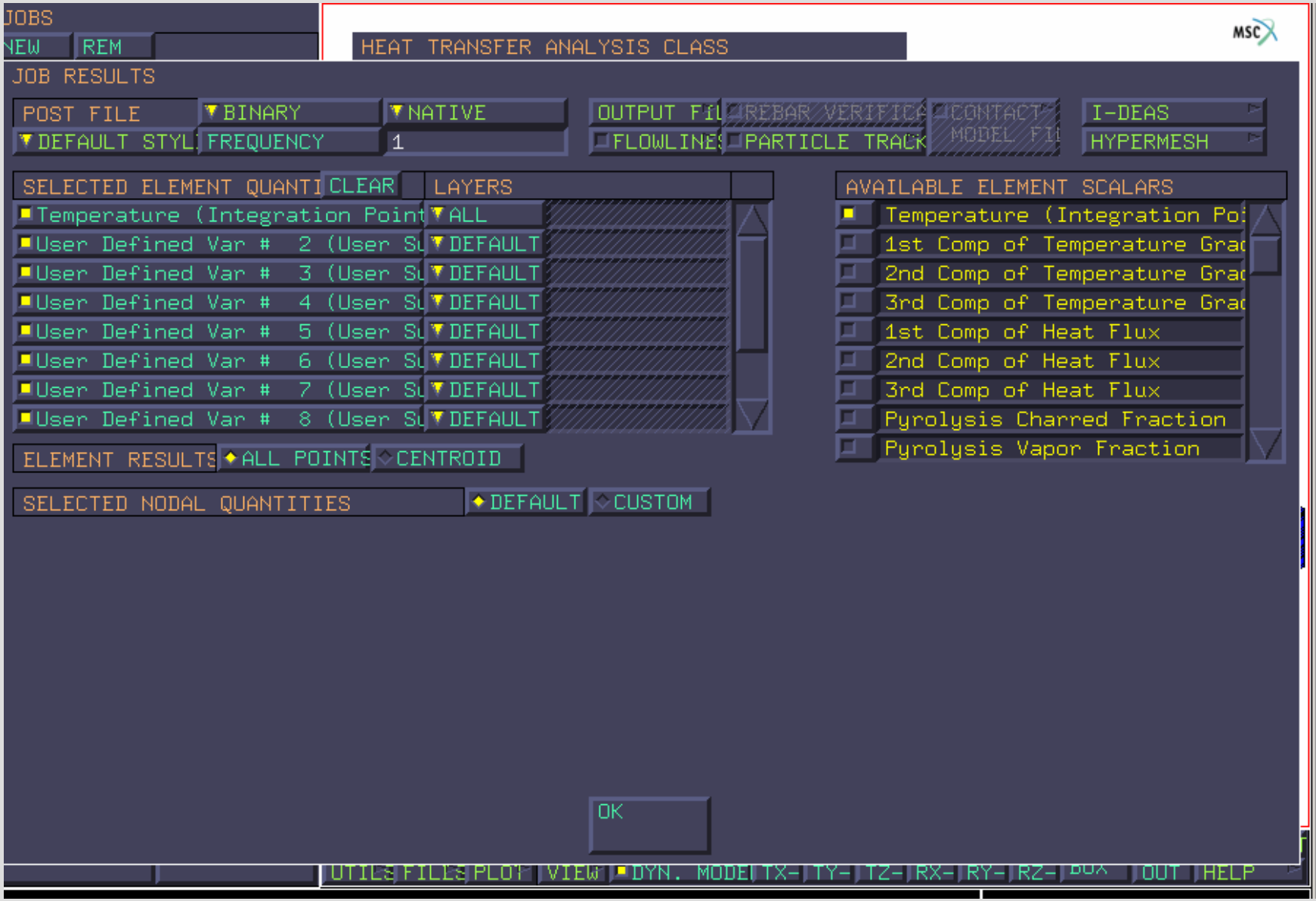
The Main Menu



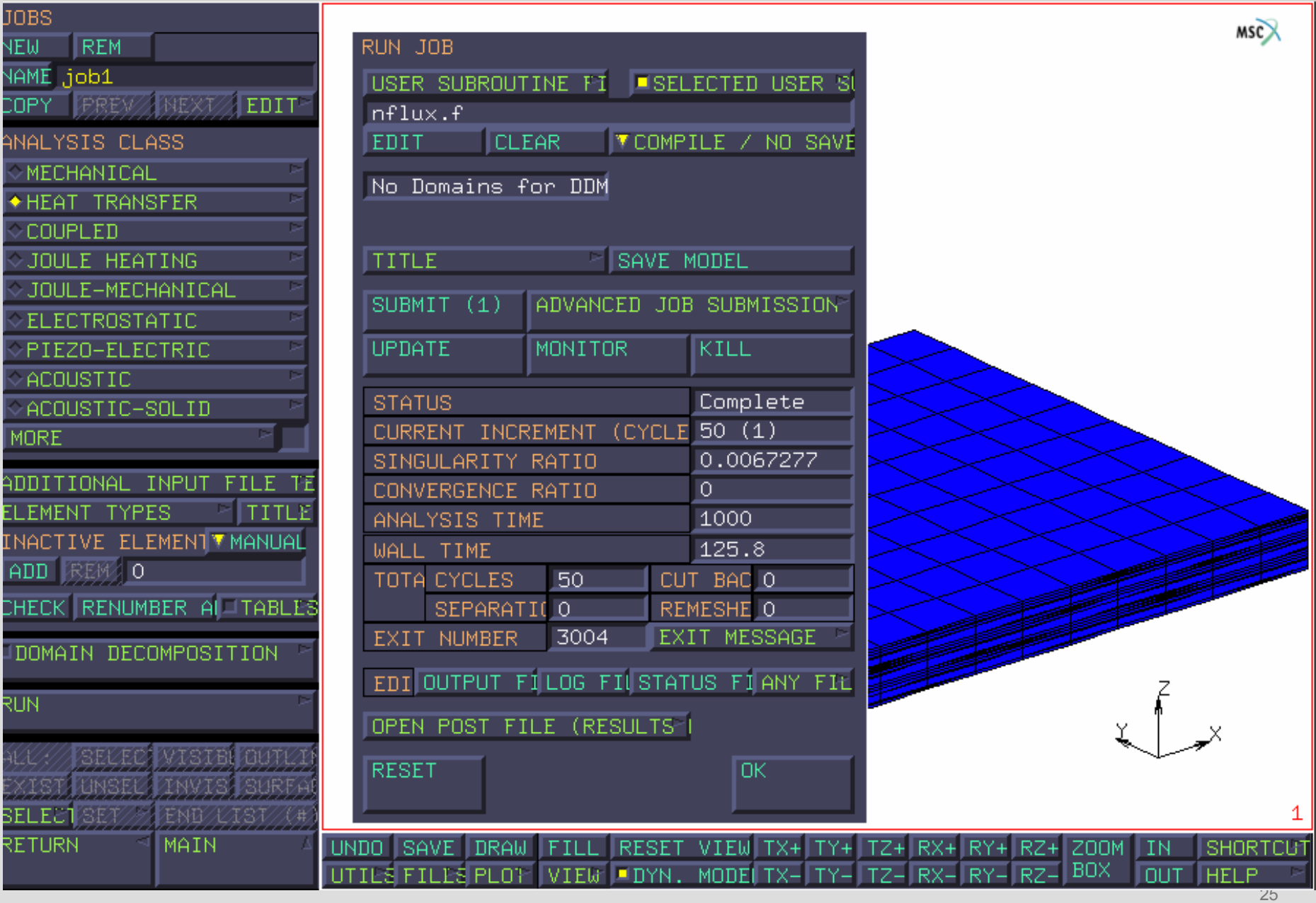
Invoking the flux user-subroutine



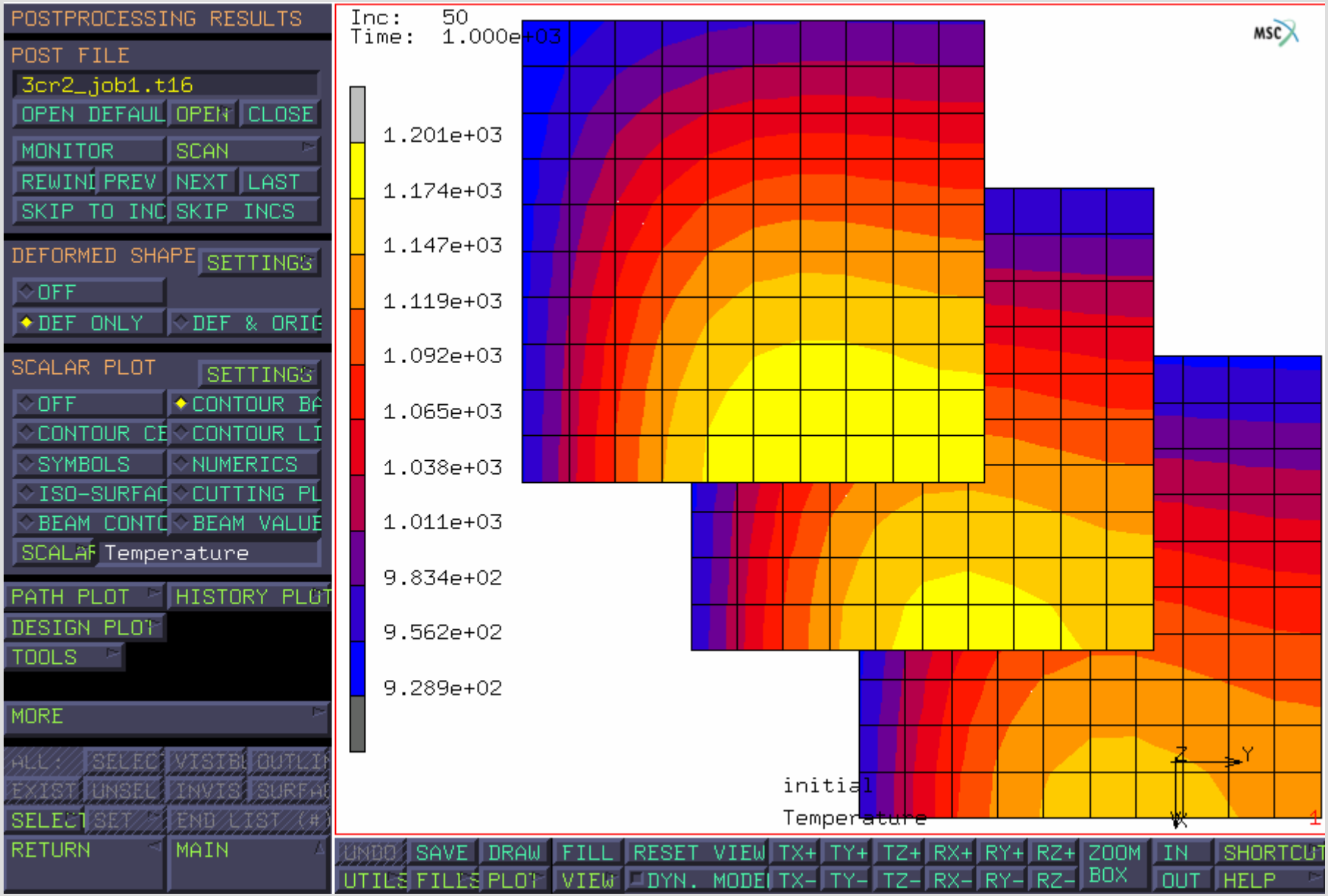
Job Results Menu - User plot variables



The Run Menu


JOBS
NEW
REM
NAME job1
COPY
PREV
NEXT
EDIT
ANALYSIS CLASS
MECHANICAL
HEAT TRANSFER
COUPLED
JOULE HEATING
JOULE-MECHANICAL
ELECTROSTATIC
PIEZO-ELECTRIC
ACOUSTIC
ACOUSTIC-SOLID
MORE
ADDITIONAL INPUT FILE TE
ELEMENT TYPES
TITLE
INACTIVE ELEMENT
MANUAL
ADD
REM
0
CHECK
RENUMBER AI
TABLES
DOMAIN DECOMPOSITION
RUN
RETURN
MAIN
RUN JOB
USER SUBROUTINE FI
SELECTED USER SU
nflux.f
EDIT
CLEAR
COMPILE / NO SAVE
No Domains for DDM
TITLE
SAVE MODEL
SUBMIT (1)
ADVANCED JOB SUBMISSION
UPDATE
MONITOR
KILL
STATUS
Complete
CURRENT INCREMENT (CYCLE
50 (1)
SINGULARITY RATIO
0.0067277
CONVERGENCE RATIO
0
ANALYSIS TIME
1000
WALL TIME
125.8
TOTA
CYCLES
50
CUT BAC
0
SEPARATI
0
REMESHE
0
EXIT NUMBER
3004
EXIT MESSAGE
EDI
OUTPUT FI
LOG FI
STATUS FI
ANY FIL
OPEN POST FILE (RESULTS
RESET
OK
MSC
Z
Y
X
1
UNDO
SAVE
DRAW
FILL
RESET VIEW
TX+
TY+
TZ+
RX+
RY+
RZ+
ZOOM BOX
IN
SHORTCUT
UTILS
FILLS
PLOT
VIEW
DYN. MODE
TX-
TY-
TZ-
RX-
RY-
RZ-
OUT
HELP

The Results Menu



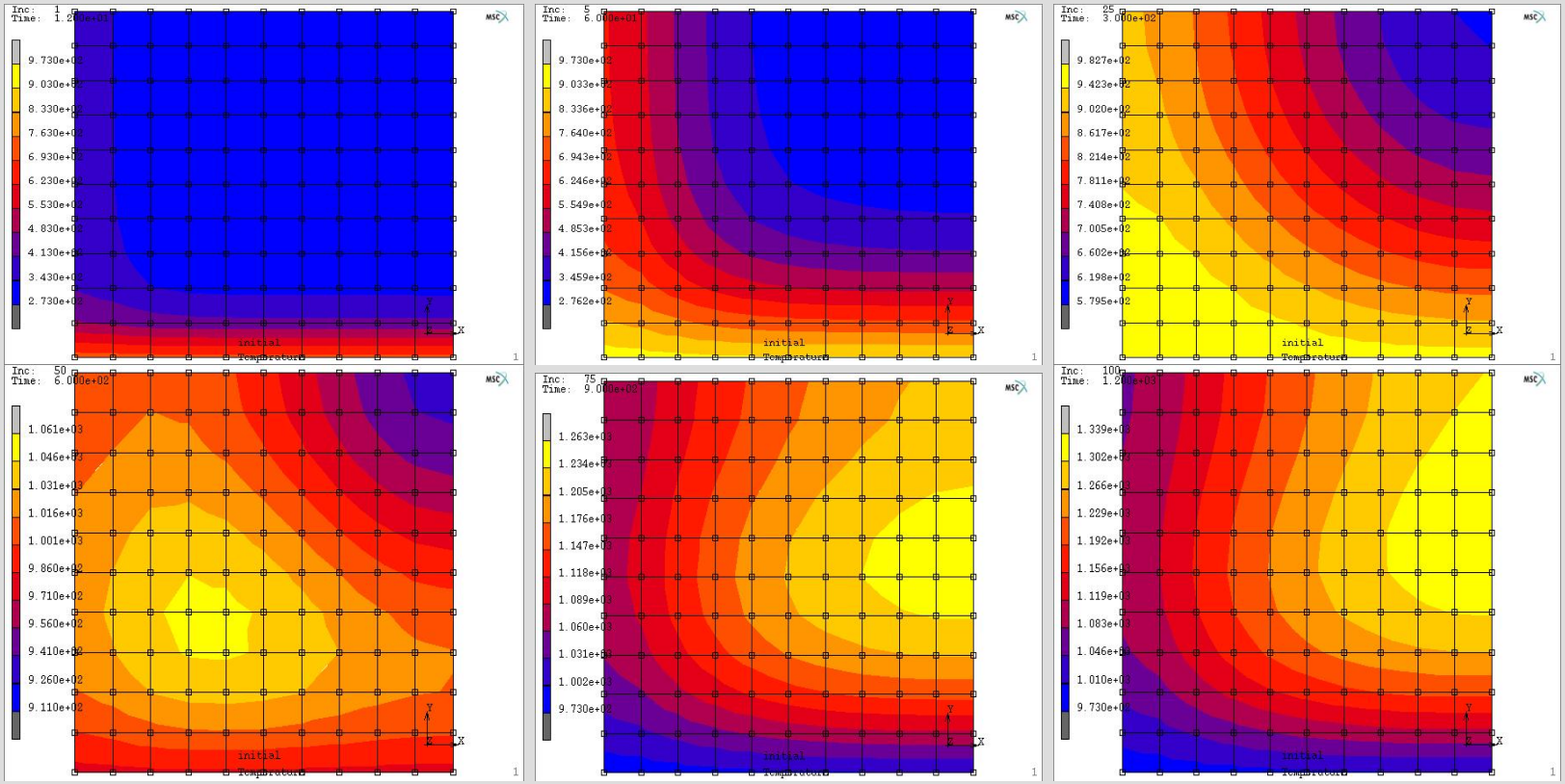
Files needed for modeling runs

1. Procedure file: generating SOFC stack model for MARC – stack geometry, material property, boundary condition, etc
2. Flux.f: user subroutine, MARC & EC module interface code
3. Cell_para.fh: property data of the experimental cell
4. Work_para.fh: data describing the working cell
5. Subroutine “plotv()”: post processing of the user defined variables

Show sample files in computer & explain

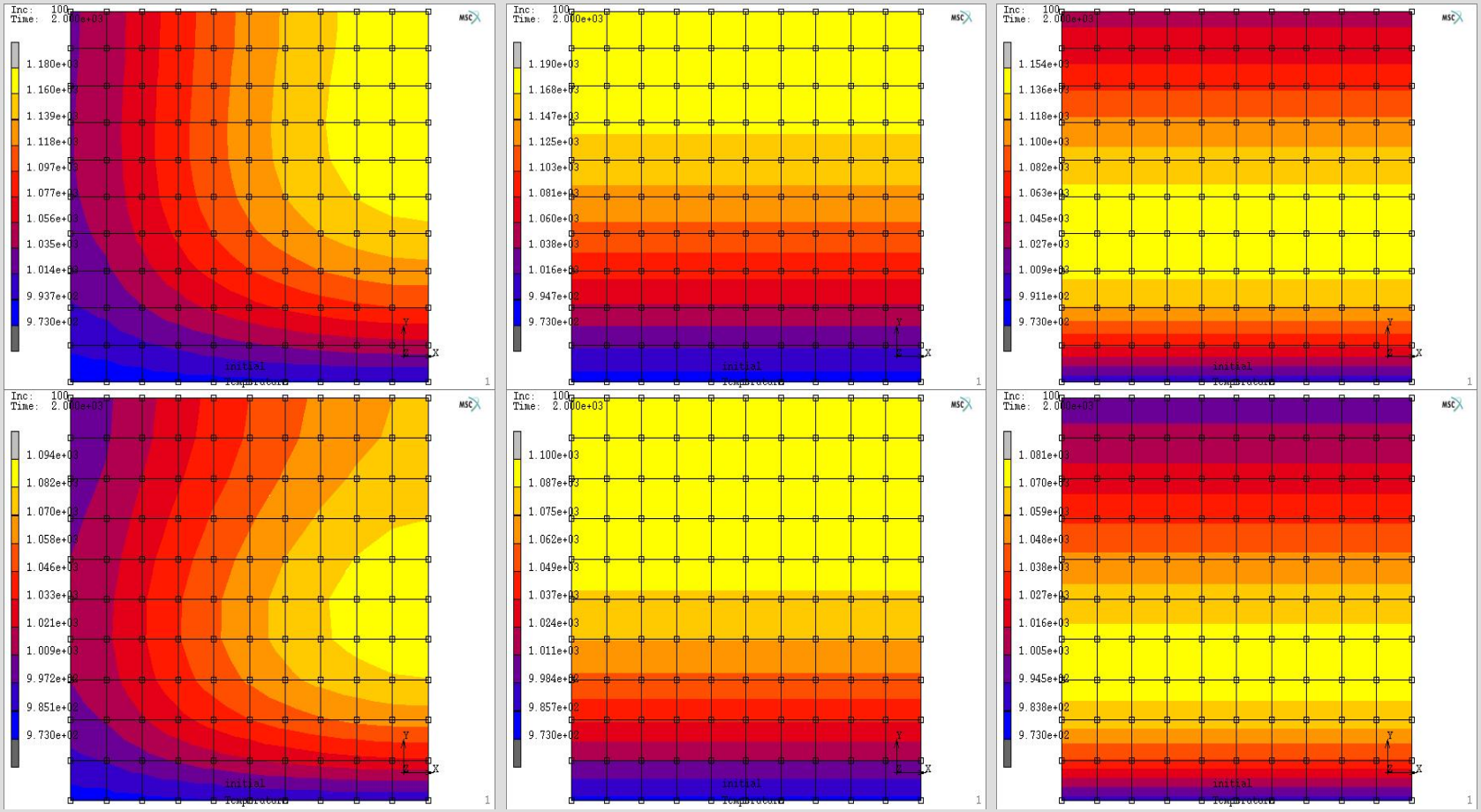
Sample results:
From startup to steady state transition

- T0=0C, Tair/fuel=700C, Vair=0.33l/s, Vfuel=0.0825l/s, V=0.7Volt
- times: a)12s; b)60s; c)300s; d)600s; e)900s; f)1200s
- Transient results almost identical for time steps from 0.2s to 12s



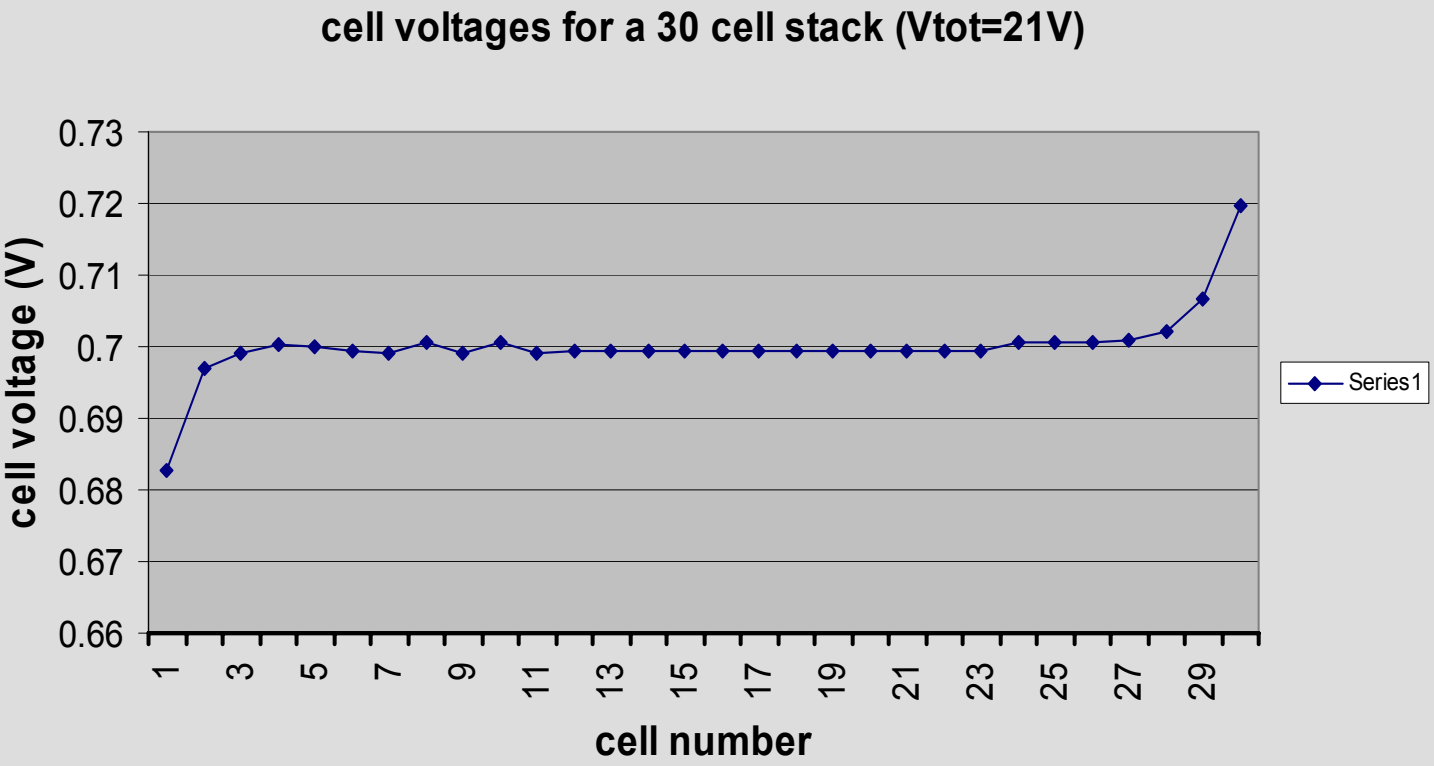
Comparison of different flow design

- Temperature profile for cross-, co-, counter-flow
- H2+CO fuel; Vfuel=0.055 & 0.0275l/s, respectively.



Sample Results for Multi-cell Stacks

- Cell voltage variation for a 30-cell stack (Vtot=21V).
- Only the outmost top and bottom 2-3 cells deviate substantially from the average; similar results for 8- & 15- cell stacks.



EC module CPU time requirement

- Table 1, CPU time when specifying total current
- # of cells 2 3 4 5 8 15 30
- Time/inc(s) 0.45 0.68 0.91 1.2 1.8 3.5 6.9

- Table 2, CPU time when specifying total voltage
- # of cells 2 3 4 5 8 15 30
- Time/inc(s) 14 27 36 51 53 149 326

- CPU time increases roughly linearly with the number of cells in the stacks.
- Typically 20-40 increments are needed for a steady state run.

Ongoing and future work

- Continued simulation of multi-cell stacks.
- Integration of the EC module for the active cell area with a general mesh of the entire fuel cell stack geometry – including the inlet and out manifolds.
- Implementation of a flow module parallel to EC
  - Greater computational efficiency.
  - Treat compressibility of flow.
  - Heat transfer in the air and fuel manifolds.

MARC-EC Tool Summary

- A multi-physics computational tool for SOFC modeling was developed with sufficient computational efficiency and numerical stability for parametric studies and data base building for use in a system model.
- Parametric studies will be performed and published to disseminate information.
- Help for using the tool will be provided as needed.