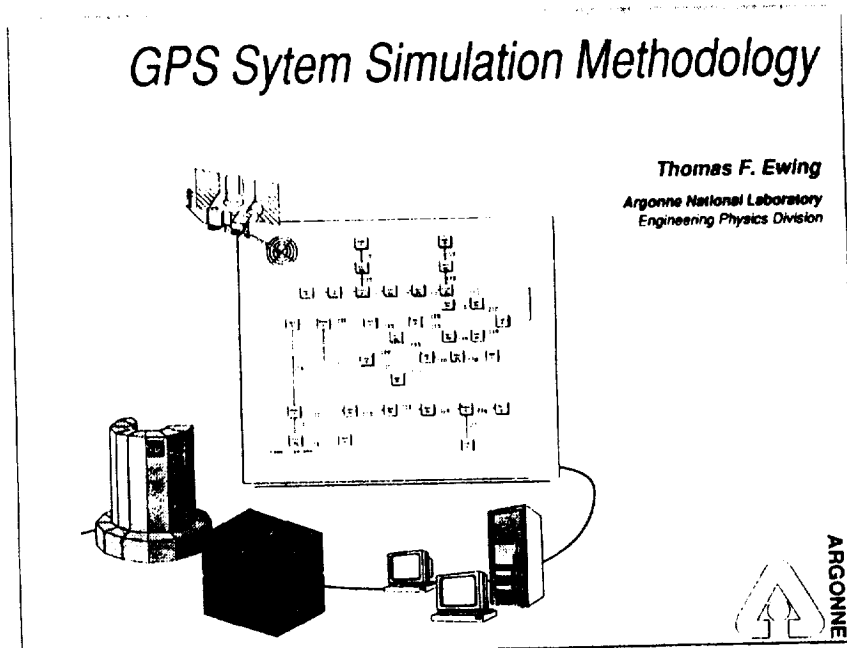


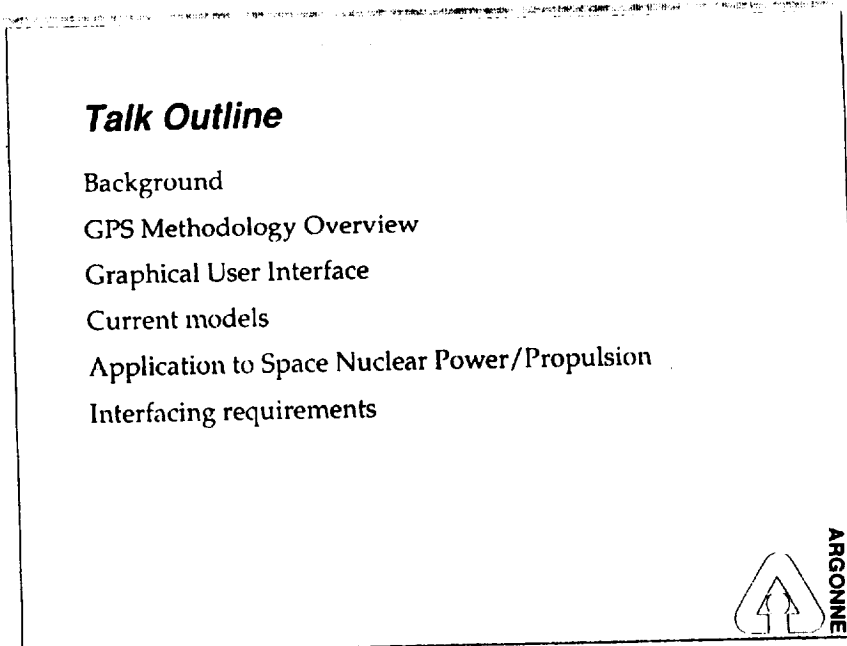
N93-26990



GPS System Simulation Methodology

Thomas F. Ewing
Argonne National Laboratory
Engineering Physics Division

T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992



Talk Outline

- Background
- GPS Methodology Overview
- Graphical User Interface
- Current models
- Application to Space Nuclear Power/Propulsion
- Interfacing requirements

T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

History

- SALT (system analysis language translator) - Early 80's
 - PL/I code for IBM mainframes
 - Moved to multiple platforms and languages (C, C++)
 - Batch oriented - translate, compile, run
 - Used model and property libraries
 - Optimizations and system analysis

Applied to

- Open-cycle and liquid-metal MHD systems
- Fuel cells
- Ocean thermal energy conversion
- Municipal solid waste processing
- Fusion
- Breeder reactors
- Geothermal and solar energy systems



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Next Generation Implementation - GPS

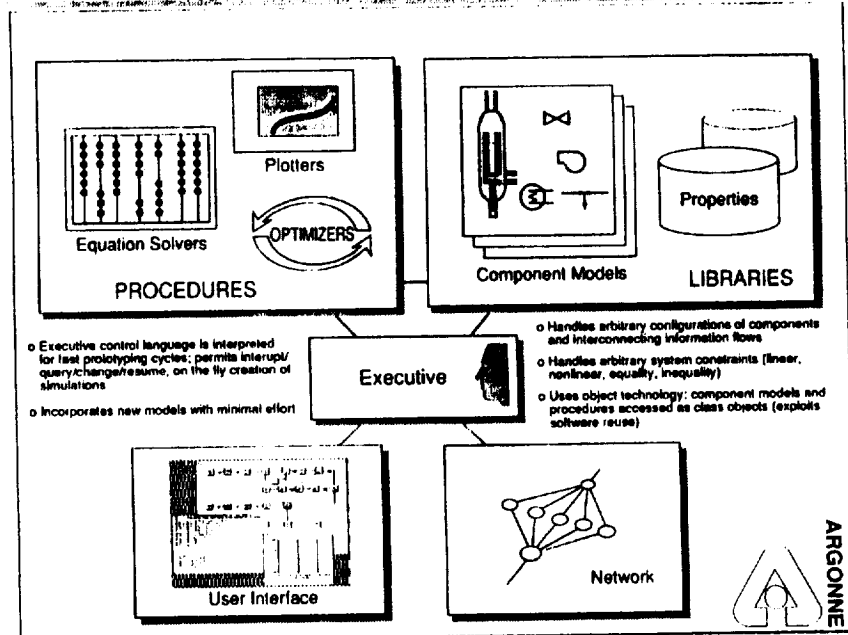
- Designed for modern workstation environments
- Developed in C++, moved to C for greater portability
- Steady-state & dynamic model libraries concept of SALT, but accessed as *class objects*
- Complete, extensible, object-oriented control language with numerous procedures for optimizations, equations solving, system constraints, parametric analysis
- Language interpreted, but uses compiled, fully optimized models and math procedures ==>
 - Fast prototyping cycles
 - On-the-fly creation of interaction with simulations
 - Simulation systems can be interrupted, queried and changed, then resumed



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Simulation/Modeling Approach



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
 Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

GPS Operators

- 86 built-in operators
- I/O functions (fopen, printf, scanf, sprintf)
- Math functions (atan2, pow, exp, max, ln, log10)
- Numerical procedures (vary, cons, icons, mini, diff)
- Looping and flow control
 - cond (...) if
 - cond (...) (...) ifelse
 - start inc bound (...) for
 - count (...) repeat
 - {...} loop
 - {cond} {...} while

T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
 Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Miscellaneous Operators

- Allocate new model class instance - **cdef**
/pump1 { pump: /param1 12.0 /param2 0.495 } cdef
- Set a debug level (0 thru 5) - **debug**
- Run gps simulation from a input file - **run**
"input.fil" run
- Interrupt simulation to permit queries/interactions
sintrp (followed by **resume** to continue)



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

GPS Steady-State Power System Models

Basic component models

gas - gas flow initiator
sp - gas flow splitter
mx - gas flow mixer
ht - gas flow heater/cooler
hx - gas flow heat exchanger
cp - compressor
gt - gas turbine
pump - pump
df - diffuser
nz - nozzle
power - calculate system powers

Basic thermionic models

react - reactor model
ti - thermionic converter
rad - thermal radiator
sp - power flow splitter
res - electrical resistor
bc - boost converter
bus - electrical bus
mass - mass calculations

More sophisticated models

therm - thermal flow initiator
hprad - heat pipe radiator
tds - thermionic diode subsystem
shx - simple, multinode heat exchanger
nhx - multinode, general purpose HT model



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

GPSTool - Graphical User Interface

Parameter panels provide access to system variables and model default values

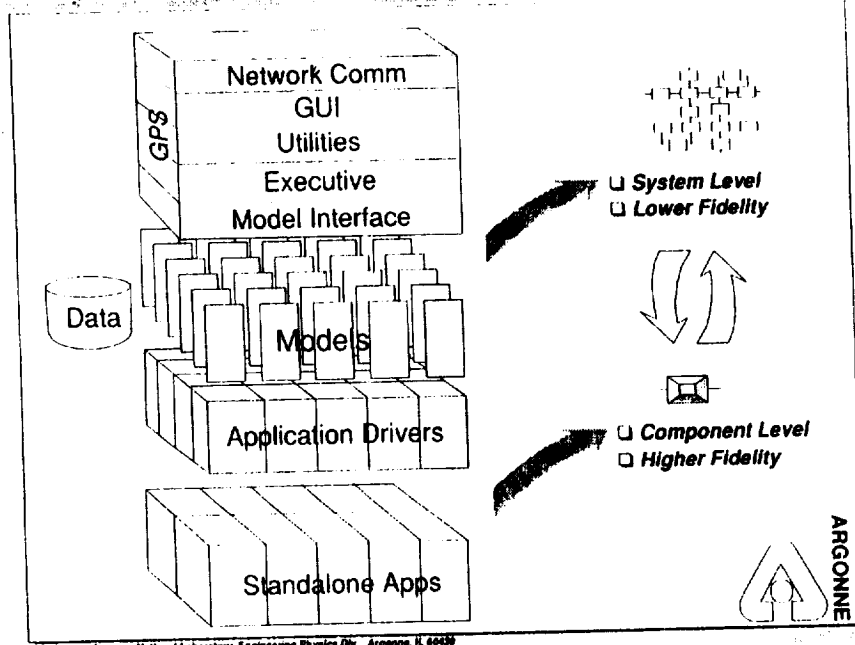
GPSTool panel provides point-and-click creation and control of simulations. Using object technology, GPS assembles simulations from standard software parts: model libraries, databases, & procedure libraries.

Supports drag-and-drop loads of input files or templates

Control language governing simulation and generated by GPSTool appears here; may be edited or manually input by power users

Graphics options includes generation of plots and system diagram. System diagram also permits mouse interactions to select components for insertion, deletion, rearrangement, parameter changes, etc.

Phillips Lab Simulation Strategy



T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
 Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Classes

- ch
- cp
- at
- lk
- gas
- geom
- gt
- ht
- hr

Model: tds_1

change value Insert parm

deno = 1.9600e-02

deno1 = 1.9600e-02
 dcl = 2.0600e-02
 dco = 2.3400e-02
 length = 3.0900e-01
 cones = -1.0000e+00
 vques = 9.2000e-01
 Item = (substructure)
 fco = (substructure)

File Manager: /tmp/mat/home/kinga/poyer/gps1001/state

GPS System Configuration

```

  graph TD
    deno1[deno1] --- deno2[deno2]
    deno1 --- deno3[deno3]
    deno2 --- deno4[deno4]
    deno3 --- deno5[deno5]
    deno4 --- deno6[deno6]
    deno5 --- deno7[deno7]
    deno6 --- deno8[deno8]
    deno7 --- deno9[deno9]
    deno8 --- deno10[deno10]
    deno9 --- deno11[deno11]
    deno10 --- deno12[deno12]
    deno11 --- deno13[deno13]
    deno12 --- deno14[deno14]
    deno13 --- deno15[deno15]
    deno14 --- deno16[deno16]
    deno15 --- deno17[deno17]
    deno16 --- deno18[deno18]
    deno17 --- deno19[deno19]
    deno18 --- deno20[deno20]
    deno19 --- deno21[deno21]
    deno20 --- deno22[deno22]
    deno21 --- deno23[deno23]
    deno22 --- deno24[deno24]
    deno23 --- deno25[deno25]
    deno24 --- deno26[deno26]
    deno25 --- deno27[deno27]
    deno26 --- deno28[deno28]
    deno27 --- deno29[deno29]
    deno28 --- deno30[deno30]
    deno29 --- deno31[deno31]
    deno30 --- deno32[deno32]
    deno31 --- deno33[deno33]
    deno32 --- deno34[deno34]
    deno33 --- deno35[deno35]
    deno34 --- deno36[deno36]
    deno35 --- deno37[deno37]
    deno36 --- deno38[deno38]
    deno37 --- deno39[deno39]
    deno38 --- deno40[deno40]
    deno39 --- deno41[deno41]
    deno40 --- deno42[deno42]
    deno41 --- deno43[deno43]
    deno42 --- deno44[deno44]
    deno43 --- deno45[deno45]
    deno44 --- deno46[deno46]
    deno45 --- deno47[deno47]
    deno46 --- deno48[deno48]
    deno47 --- deno49[deno49]
    deno48 --- deno50[deno50]
    deno49 --- deno51[deno51]
    deno50 --- deno52[deno52]
    deno51 --- deno53[deno53]
    deno52 --- deno54[deno54]
    deno53 --- deno55[deno55]
    deno54 --- deno56[deno56]
    deno55 --- deno57[deno57]
    deno56 --- deno58[deno58]
    deno57 --- deno59[deno59]
    deno58 --- deno60[deno60]
    deno59 --- deno61[deno61]
    deno60 --- deno62[deno62]
    deno61 --- deno63[deno63]
    deno62 --- deno64[deno64]
    deno63 --- deno65[deno65]
    deno64 --- deno66[deno66]
    deno65 --- deno67[deno67]
    deno66 --- deno68[deno68]
    deno67 --- deno69[deno69]
    deno68 --- deno70[deno70]
    deno69 --- deno71[deno71]
    deno70 --- deno72[deno72]
    deno71 --- deno73[deno73]
    deno72 --- deno74[deno74]
    deno73 --- deno75[deno75]
    deno74 --- deno76[deno76]
    deno75 --- deno77[deno77]
    deno76 --- deno78[deno78]
    deno77 --- deno79[deno79]
    deno78 --- deno80[deno80]
    deno79 --- deno81[deno81]
    deno80 --- deno82[deno82]
    deno81 --- deno83[deno83]
    deno82 --- deno84[deno84]
    deno83 --- deno85[deno85]
    deno84 --- deno86[deno86]
    deno85 --- deno87[deno87]
    deno86 --- deno88[deno88]
    deno87 --- deno89[deno89]
    deno88 --- deno90[deno90]
    deno89 --- deno91[deno91]
    deno90 --- deno92[deno92]
    deno91 --- deno93[deno93]
    deno92 --- deno94[deno94]
    deno93 --- deno95[deno95]
    deno94 --- deno96[deno96]
    deno95 --- deno97[deno97]
    deno96 --- deno98[deno98]
    deno97 --- deno99[deno99]
    deno98 --- deno100[deno100]
  
```

T. P. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
 Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Classes

- ch
- cp
- at
- lk
- gas
- geom
- gt
- ht
- hr

Model: tds_1

change value Insert parm

deno = 1.9600e-02

deno1 = 1.9600e-02
 dcl = 2.0600e-02
 dco = 2.3400e-02
 length = 3.0900e-01
 cones = -1.0000e+00
 vques = 9.2000e-01
 Item = (substructure)
 fco = (substructure)

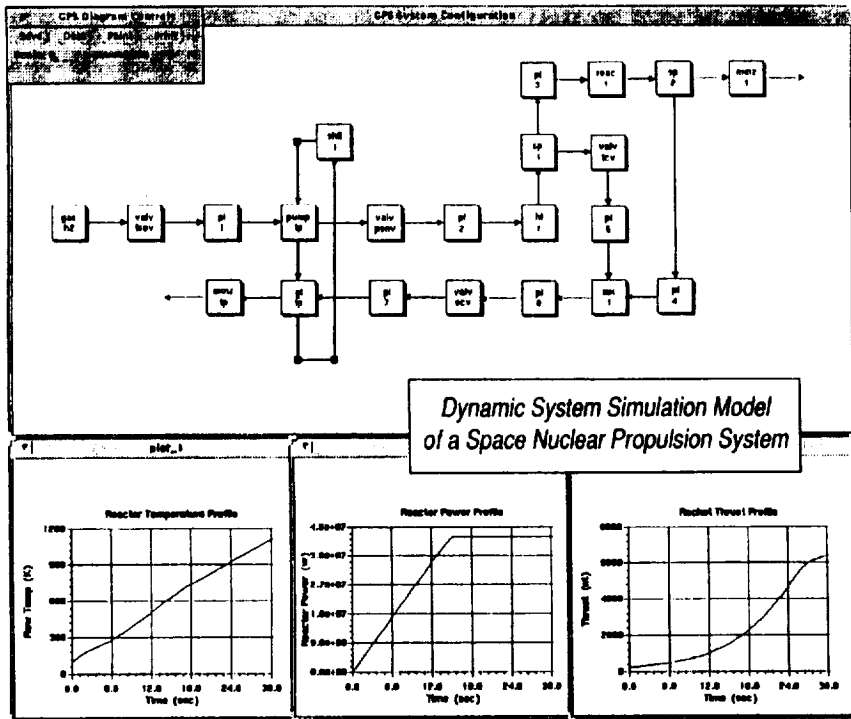
File Manager: /tmp/mat/home/kinga/poyer/gps1001/state

GPS System Configuration

```

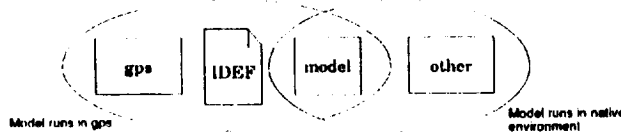
  graph TD
    deno1[deno1] --- deno2[deno2]
    deno1 --- deno3[deno3]
    deno2 --- deno4[deno4]
    deno3 --- deno5[deno5]
    deno4 --- deno6[deno6]
    deno5 --- deno7[deno7]
    deno6 --- deno8[deno8]
    deno7 --- deno9[deno9]
    deno8 --- deno10[deno10]
    deno9 --- deno11[deno11]
    deno10 --- deno12[deno12]
    deno11 --- deno13[deno13]
    deno12 --- deno14[deno14]
    deno13 --- deno15[deno15]
    deno14 --- deno16[deno16]
    deno15 --- deno17[deno17]
    deno16 --- deno18[deno18]
    deno17 --- deno19[deno19]
    deno18 --- deno20[deno20]
    deno19 --- deno21[deno21]
    deno20 --- deno22[deno22]
    deno21 --- deno23[deno23]
    deno22 --- deno24[deno24]
    deno23 --- deno25[deno25]
    deno24 --- deno26[deno26]
    deno25 --- deno27[deno27]
    deno26 --- deno28[deno28]
    deno27 --- deno29[deno29]
    deno28 --- deno30[deno30]
    deno29 --- deno31[deno31]
    deno30 --- deno32[deno32]
    deno31 --- deno33[deno33]
    deno32 --- deno34[deno34]
    deno33 --- deno35[deno35]
    deno34 --- deno36[deno36]
    deno35 --- deno37[deno37]
    deno36 --- deno38[deno38]
    deno37 --- deno39[deno39]
    deno38 --- deno40[deno40]
    deno39 --- deno41[deno41]
    deno40 --- deno42[deno42]
    deno41 --- deno43[deno43]
    deno42 --- deno44[deno44]
    deno43 --- deno45[deno45]
    deno44 --- deno46[deno46]
    deno45 --- deno47[deno47]
    deno46 --- deno48[deno48]
    deno47 --- deno49[deno49]
    deno48 --- deno50[deno50]
    deno49 --- deno51[deno51]
    deno50 --- deno52[deno52]
    deno51 --- deno53[deno53]
    deno52 --- deno54[deno54]
    deno53 --- deno55[deno55]
    deno54 --- deno56[deno56]
    deno55 --- deno57[deno57]
    deno56 --- deno58[deno58]
    deno57 --- deno59[deno59]
    deno58 --- deno60[deno60]
    deno59 --- deno61[deno61]
    deno60 --- deno62[deno62]
    deno61 --- deno63[deno63]
    deno62 --- deno64[deno64]
    deno63 --- deno65[deno65]
    deno64 --- deno66[deno66]
    deno65 --- deno67[deno67]
    deno66 --- deno68[deno68]
    deno67 --- deno69[deno69]
    deno68 --- deno70[deno70]
    deno69 --- deno71[deno71]
    deno70 --- deno72[deno72]
    deno71 --- deno73[deno73]
    deno72 --- deno74[deno74]
    deno73 --- deno75[deno75]
    deno74 --- deno76[deno76]
    deno75 --- deno77[deno77]
    deno76 --- deno78[deno78]
    deno77 --- deno79[deno79]
    deno78 --- deno80[deno80]
    deno79 --- deno81[deno81]
    deno80 --- deno82[deno82]
    deno81 --- deno83[deno83]
    deno82 --- deno84[deno84]
    deno83 --- deno85[deno85]
    deno84 --- deno86[deno86]
    deno85 --- deno87[deno87]
    deno86 --- deno88[deno88]
    deno87 --- deno89[deno89]
    deno88 --- deno90[deno90]
    deno89 --- deno91[deno91]
    deno90 --- deno92[deno92]
    deno91 --- deno93[deno93]
    deno92 --- deno94[deno94]
    deno93 --- deno95[deno95]
    deno94 --- deno96[deno96]
    deno95 --- deno97[deno97]
    deno96 --- deno98[deno98]
    deno97 --- deno99[deno99]
    deno98 --- deno100[deno100]
  
```

T. P. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439
 Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992



Advantages as Integrating Environment

- Consistent user interface to models
- Diverse models can be combined for use in arbitrarily complex systems
- Suite of gps system analysis capabilities (sweeps, optimizations) and numerical methods/properties available to models
- Interface definitions external to models ==>
 - can adapt models developed independent of gps
 - can use proprietary models available only as object code
 - models used with gps can still be run in native mode



V. F. Emig, Argonne National Laboratory, Engineering Physics Div., Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting, NASA Lewis Research Center, October 20-23, 1992

Interfacing Considerations

- Component models can be Fortran, C, or other Sun languages which generate linkable object code
- Standalone codes must be structured as subroutines with argument list of variables/parameters that must be known to GPS system
- Use of Fortran common blocks prevents (presently) having multiple instances of that model in a system
- Because models may be cycled through numerous convergence iterations with perturbed input flows

Models must be true functions of their inputs

Models must be reasonably robust

I/O routines should be moved outside computation routines



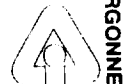
ARGONNE

T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Converting a standalone code

- Two step process:
 - Convert code to one or more subroutines
 - Create a interface definition file (IDEF)
- GPS uses IDEF to generate small C code to handle interfaces
- Model can still be run independently of gps (standalone) by writing a main program to call subroutine



ARGONNE

T. F. Ewing Argonne National Laboratory Engineering Physics Div. Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992

Interface Specification File Format

Interface specifications external to models

- User-prepared ASCII file used by GPS preprocessor to generate C stub code to handle gps interfacing
 - Model name
 - Variable types and initial values (arguments + gps I/O)
 - Entry procedures (name, arguments if Fortran routine, in and out flow variables)
 - Print variables (used as default gps output)

<p>EXAMPLE Thermionic reac model</p> <p>c - mass/sizing, thermionic component of power g - waste heat flow</p> <p><small>T. F. Ewing Argonne National Laboratory Nuclear Propulsion Technical I</small></p>	<pre> model: reac char: names[16] names[16] names[16] nameboom[16] double: pow = 1e6 eff = 0.13 radius height sep = 10.0 rhoboom = 10.0 lboom radiusr vols heights = 0.37 flowtype: ll lls masstype: mcore mss mrs mboom entry: c outflow: mcore mss mrs mboom ll entry: s outflow: lls print: pow eff radius height print: radiusr vols heights sep </pre>	<p>INTERFACE SPECIFICATION</p> <p>ARGONNE</p>
--	---	---

Example Conversion

Fortran Standalone code - TDS

- 8400 lines of Fortran code (includes TECMDL)
- Required 32 line interface definition file
- Conversion completed in < 2 hrs.
- Same model now runs standalone (called from main) or in gps environment
- Both open (once through) and closed systems have been run in gps
- Have successfully run problems with 250,000 nonlinear constraints in nested loops



T. F. Ewing Argonne National Laboratory Engineering Physics Div Argonne, IL 60439

Nuclear Propulsion Technical Interchange Meeting NASA-Lewis Research Center October 20-23, 1992