



HSPF Conversion Project

Presentation to Edwards Aquifer Authority
Russell Persyn, Paul Hummel, and Jason Love
January 2017



Outline

- **Hydrologic Simulation Program–Fortran (HSPF)**
 - Overview
 - Current Challenges
- **The Hydrologic Simulation Program–Python (HSP²) Conversion Project**
 - Project Goals
 - HSP² Modeling System
- **Sharing Data and Reproducible Analysis with Jupyter Notebooks and Python Pandas Library**
- **Questions**



HSPF Background

- **Hydrologic Simulation Program–FORTRAN (HSPF)**
 - Public domain
 - Strong model documentation, developer/sponsor assistance, user's group, workshops, code control
 - Continuous Simulation Model
 - Represents complex multi-land use watersheds
 - Land surface and subsurface hydrology and water-quality processes
 - Stream and lake hydraulics and water-quality
 - BMP module to facilitate implementation planning



**US ARMY
CORPS OF ENGINEERS**



Current Challenges with HSPF Model

- **Complexity and Rigidity of Fortran-77 Memory Management has Hindered Maintenance and Upgrading Functionality**
 - Engineering code is intertwined with and complicated by memory management code.
 - Aspects of managing memory and Fortran-77 Common Block are undocumented (e.g., SEQ file).
 - Model size and number of operations is limited.
 - The land use, translation factors, and many parameters are fixed.
 - In some cases, the complicated Special Actions Module can be used to mimic these real-world varying conditions but the complexity of this module significantly increases the time to develop the input file and slows down the model execution times.



Current Challenges with HSPF Model

- **Legacy Code and Data Model Limited in Ability to Integrate with Modern Software and Leverage Parallel Computing.**
 - The ASCII text “Punch Card” style UCI is difficult for modern software to interact with.
 - WDM 32-bit architecture limits the size of WDM file.
 - WDM is not supported by any Commercial Off-the-Shelf software making pre/post-processing of time series data cumbersome.
 - WDM is limited in the type of data/files it can store.
 - Code is not MPI enabled to run parallel on a cluster.
- **Voids in Pre-Processing, Optimization, and Post-Processing Tools**



Project Goal and Objectives

- **GOAL:** Mitigate the aforementioned challenges currently facing the HSPF model so it will continue to be relevant into the foreseeable future.
 - **Retain all current functionality**, from the user's point of view, and **provide a migration path** for legacy applications.
 - Provide documentation within new code to **transparently show the translation path**.
 - **Elevate engineering code** to make the engineering/science clear, not lost within the memory management aspects of the code.
 - Restructure for maintainability, to **remove fixed limits (e.g., operations, land use, parameters)**, and to maintain or improve execution time.
 - Code should be **independent of operating system and hardware**.
 - New code should be **compatible with multiple cores and GPU** for acceleration.
 - Place code in **open source** and freely distributed over the web.
 - Facilitate **Sharing Data and Reproducible Analysis**.



Solution

- **Convert Code to a Modern Widely Accepted, Open Source, High Performance Computing (HPC) Code**

- Python and HPC Packages



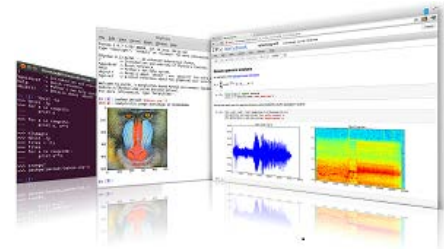
- **Convert Model Input /Output Files to Modern Widely Accepted, Open Source, Data Model, Library, and Binary File Format**

- HDF5



- **Package to Facilitate Organization, Documentation, and Collaboration**

- Jupyter Notebook



Why Python?



- **Clean And Simple Language**
 - Easy-to-read and intuitive code, easy-to-learn minimalistic syntax, maintainability scales well with size of projects
- **Expressive Language**
 - Fewer lines of code, fewer bugs, easier to maintain.
- **Dynamically Typed**
 - No need to define type of variables, function arguments or return types.
- **Automatic Memory Management**
 - No need to explicitly allocate and deallocate memory for variables and data arrays.
- **Interpreted**
 - No need to compile the code. The Python interpreter reads and executes the python code directly.



Why Python?

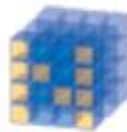
- Readily Available Open Source Solutions

```
from numba import jit
from numpy import arange

# jit decorator tells Numba to compile this function.
# The argument types will be inferred by Numba when the function is called.
@jit
def sum2d(arr):
    M, N = arr.shape
    result = 0.0
    for i in range(M):
        for j in range(N):
            result += arr[i,j]
    return result

a = arange(9).reshape(3,3)
print(sum2d(a))
```

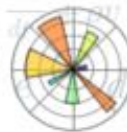
SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:



NumPy
Base N-dimensional
array package



SciPy library
Fundamental
library for scientific
computing



Matplotlib
Comprehensive 2D
Plotting

IP[y]:
IPython

IPython
Enhanced
Interactive Console



Sympy
Symbolic
mathematics



pandas
Data structures &
analysis



Why Python?

- **Ipython/Jupyter Notebooks**

- Server-Client Application
- Runs in a Web-Browser for remote or local hosting
- Kernel is the “computational engine”
 - Can mix languages
- Interactive code
 - Introspection
- Output controls
 - Widgets
- Shareable documents

Notebooks - contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, media files, etc...).

The screenshot displays the IPython Interactive Computing website. At the top, the logo 'IP[y]: IPython Interactive Computing' is prominent. Below it, the 'jupyter nbviewer' logo is visible on the left, and 'JUPYTER' and 'FAQ' links are on the right. The main content area is a grid of nine featured notebooks:

- Data Visualization with Lightning:** Shows various plots including a scatter plot, a line plot, and a heatmap.
- Interactive data visualization with Bokeh:** Displays a collection of interactive plots, including a scatter plot, a line plot, and a heatmap.
- Interactive plots with Plotly:** Shows a grid of various interactive plots, including a scatter plot, a line plot, and a heatmap.
- XKCD Plot With Matplotlib:** Features a plot titled 'XKCD plots in Matplotlib' showing a sine wave and a cosine wave.
- Python for Vision Research:** Displays the Python logo and the 'CISTAL REVISION' logo.
- Non Parametric Regression:** Shows a plot titled 'Covariance function' with a graph of a function and its covariance matrix.
- Partial Differential Equations Solver:** Features a plot titled 'The Crank-Nicolson Method' showing a 2D heatmap of a solution.
- Analysis of current events:** Displays a scatter plot of data points.
- Jaynes-Cummings model:** Shows a plot of the Jaynes-Cummings model with various parameters and a graph of the resulting wave function.



Why HDF5?

- **HDF5 supports all types of data stored digitally, regardless of origin, size of complexity.**
 - **Petabytes** of remote sensing and GIS data collected by satellites
 - **Terabytes** of computational results from nuclear testing models
 - **Megabytes** of high-resolution MRI brain scans
 - **Metadata** necessary for efficient data sharing, processing, visualization, and archiving
- **Designed for flexible and efficient I/O and for high volume.**
- **Open source tools and applications for managing, manipulating, viewing, and analyzing the stored data are readily available.**



HSP² Modeling System

HSP² Development Process

- **Tools to illuminate the HSPF code prior to code conversion**
 - Number and reformat Fortran code to make determining loop and branch limits more accurate
 - Draw a network diagram of the Fortran code call graph
- **Tool to show traceability between HSPF and HSP²**
 - HSP² has line numbers of corresponding HSPF code as comments
 - Module names are also traced between HSPF and HSP² equivalent code
 - Variable names are basically unchanged for traceability
 - Many comments from the HSPF code have been carried to the HSP² code
- **Verification Tools**
 - Rapidly compare the results of HSPF verification tests with HSP² results
 - HDFView to check results of data storage operations
- **Version Control**
 - GitHub



HSP² Modeling System

Code Example

Lines of Code
Typically 5 – 10 times less

IWATER - HSP2 (n=234)
HSPF = (n=1027)

```
284 if CSNOFG: #974
285     """ snow is being considered - allow for it find the moisture supplied
286     to interception storage. rainf is rainfall in inches/ivl. adjust for
287     fraction of land segment covered by snow. wyield is the water yielded by
288     the snowpack in inches/ivl. it has already been adjusted to an effective
289     yield over the entire land segment """
290
291     airtmpV = typeV(ts['AIRTMP']) #983-995
292     rainfV = typeV(ts['rainf']) #1005
293     snocovV = typeV(ts['snocov']) #1010
294     wyieldV = typeV(ts['wyield']) #1015
295
296     supyV = rainfV * (1.0 - snocovV) + wyieldV #1030
297
298     petadjV = (1.0 - forest) * (1.0 - snocovV) + forest #1037
299     petadjV[(airtmpV < petmax) & (petadjV > 0.5)] = 0.5 #1039, 1045-1048
300     petadjV[airtmpV < petmin] = 0.0 #1041-1044
301     ts['petadj'] = petadjV
```



HSP² Modeling System

Data Model (HDF5)

HDFView 2.9

File Window Tools Help

Recent Files C:\Project\HSP2\Tutorial2\test10.h5

test10.h5

- CONTROL
 - CONFIGURATION
 - EXT_SOURCES
 - GLOBAL
 - MASS_LINK
 - NETWORK
 - OP_SEQUENCE
 - SCHEMATIC
 - table
- FTABLES
- IMPLND
- PERLND
- REACHES
- RESULTS
 - IMPLND1
 - PERLND1
 - REACHES1
 - table
 - REACHES2
 - REACHES3
 - REACHES4
 - REACHES5
- RUN_LOG
- Timeseries
 - TS122
 - TS123
 - TS125
 - TS126
 - TS131
 - TS135
 - TS139
 - TS41

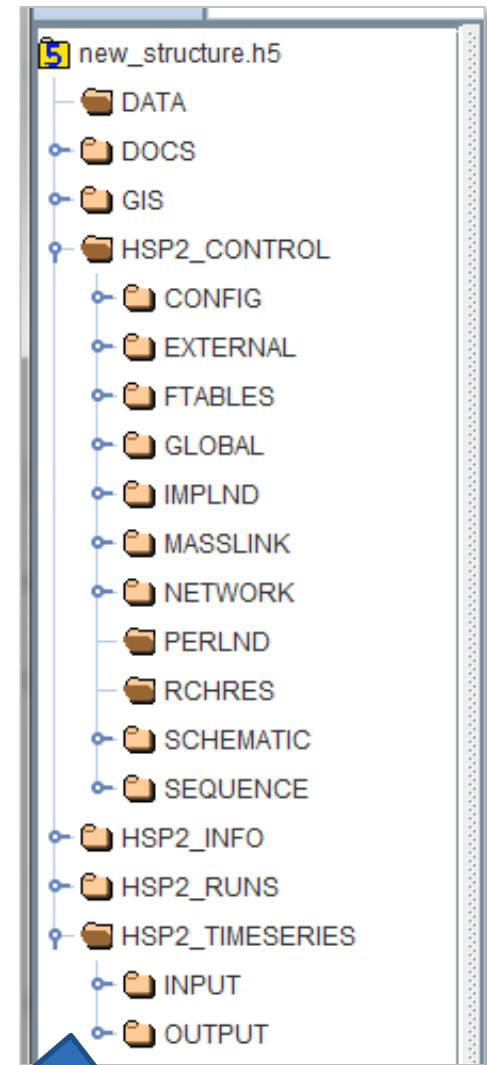
TableView - table - /CONTROL/SCHEMATIC/ - C:\Project\HSP2\Tutorial2\test10.h5

	index	Source	Source_ID	Factor	Target	Target_ID	Mass_Link	UserFilter1
0	0	PERLND	1	6000.0	REACHES	1	1	filter strings
1	1	IMPLND	1	3000.0	REACHES	5	2	filter strings
2	2	REACHES	1	1.0	REACHES	2	3	filter strings
3	3	REACHES	1	1.0	REACHES	3	4	filter strings
4	4	REACHES	2	1.0	REACHES	4	5	filter strings
5	5	REACHES	3	1.0	REACHES	4	5	filter strings
6	6	REACHES	4	1.0	REACHES	5	5	filter strings

TableView - table - /RESULTS/REACHES1/ - C:\Project\HSP2\Tutorial2\test10.h5

	index	tau	ustar	sarea	rovol	vol	o1	o2	ivol
0	189302400	0.0	0.0	6.0326085	0.0	30.021832	0.0	0.0	0.0221244
1	189306000	0.0	0.0	6.0347757	0.0	30.043402	0.0	0.0	0.0218635
2	189309600	0.0	0.0	6.0369406	0.0	30.064953	0.0	0.0	0.0218446
3	189313200	0.0	0.0	6.0391026	0.0	30.086485	0.0	0.0	0.0218258
4	189316800	0.0	0.0	6.041262	0.0	30.107998	0.0	0.0	0.0218070
5	189320400	0.0	0.0	6.043419	0.0	30.129494	0.0	0.0	0.0217882
6	189324000	0.0	0.0	6.0455728	0.0	30.150969	0.0	0.0	0.0217694
7	189327600	0.0	0.0	6.0477242	0.0	30.172426	0.0	0.0	0.0217506
8	189331200	0.0	0.0	6.049873	0.0	30.193863	0.0	0.0	0.0217319
9	189334800	0.0	0.0	6.052019	0.0	30.215282	0.0	0.0	0.0217131
10	189338400	0.0	0.0	6.0541625	0.0	30.236683	0.0	0.0	0.0216944
11	189342000	0.0	0.0	6.056303	0.0	30.258064	0.0	0.0	0.0216757
12	189345600	0.0	0.0	6.0584416	0.0	30.279427	0.0	0.0	0.0216570
13	189349200	0.0	0.0	6.060577	0.0	30.300772	0.0	0.0	0.0216384
14	189352800	2.6959322	3.7298394	6.062683	2.6088525	30.321835	0.0063134	0.0	0.0216197
15	189356400	1.3217652	8.258721E-6	6.0647287	8.3893945	30.342302	0.0139889	0.0	0.0216011
16	189360000	3.0958708	1.2639417	6.0667095	0.0014633	30.362127	0.0214232	0.0	0.0215824
17	189363600	5.51964E-10	1.6876835	6.0686274	0.0020680	30.381329	0.0286237	0.0	0.0215638
18	189367200	8.5263274	2.0975709	6.070484	0.0026537	30.399925	0.03559762	0.0	0.0215452
19	189370800	1.205432E-9	2.4940606	6.072282	0.0032210	30.417934	0.04235175	0.0	0.0215267
20	189374400	1.6046763	2.8775929	6.074023	0.0037704	30.435377	0.0488928	0.0	0.0215081
21	189378000	2.045123E-9	3.2485932	6.075708	0.0043024	30.45227	0.0552274	0.0	0.0214896
22	189381600	2.5219393	3.6074718	6.077339	0.0048177	30.468628	0.0613618	0.0	0.0214710

table (36352, 2)
Compound/d/vdata, 7
Log Info Metadata



RESPEC

HSP² Modeling System

**Library of Tutorials Developed
to Demonstrate Functionality!**

Summary of New Features

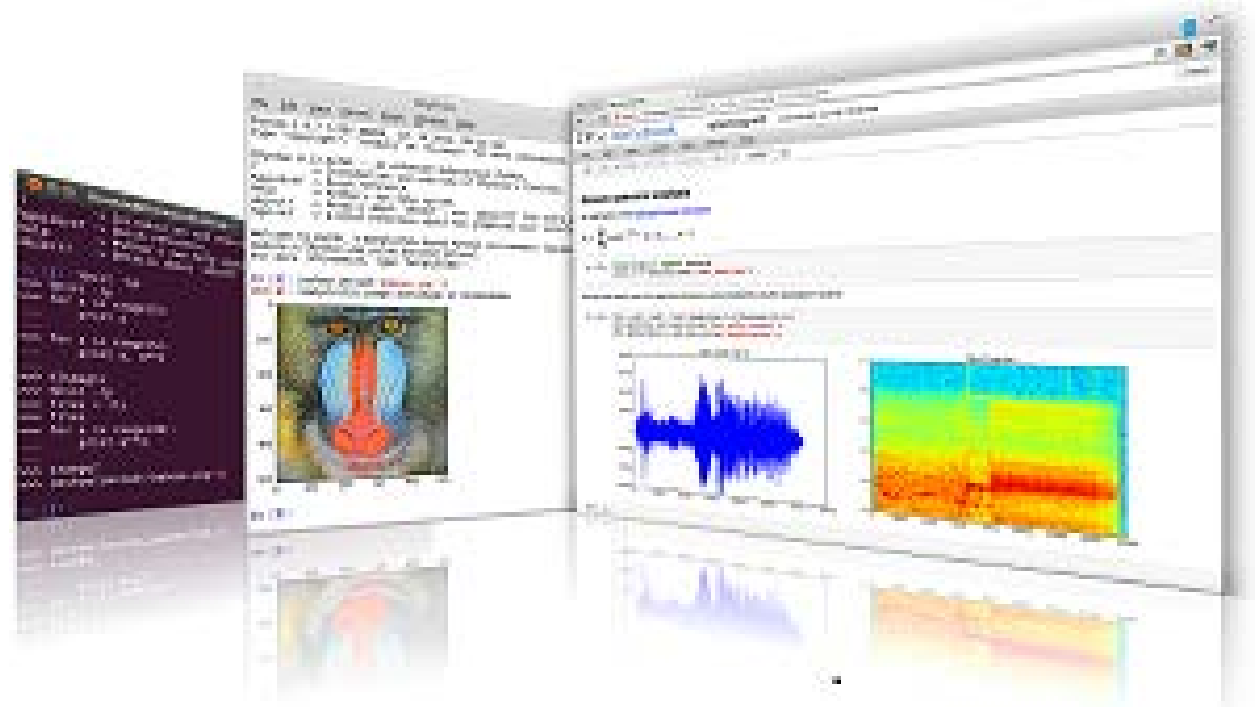
- **What was Static is now Dynamic**
 - Parameters (e.g., ASPECT, SHADE, CEPSC, LZETP)
 - Areas (e.g., ag practices, urbanization)
 - FTABLES (e.g., Geometry and Releases)
 - Other conditional changes (e.g., Water Rights, BMPs and efficiencies)
- **Easy to Add New Modules**
- **Smart Run (Efficient Optimization and Scenario Analysis)**
 - HSP² and HDF5 technology can store all necessary timeseries in the HDF5 file avoiding the need to re-run entire model application
 - Only simulate what's necessary based on changing parameters, forcing functions, and intelligent connectivity.



HSP² within Jupyter Demo

HSP² 

Jupyter





Thank You

Jason Love

Jason.Love@respec.com

RESPEC

