HSPF Conversion Project

Presentation to Edwards Aquifer Authority

Russell Persyn, Paul Hummel, and Jason Love

January 2017

RESPEC

Outline

- **Hydrologic Simulation Program–Fortran (HSPF)**
  - Overview
  - Current Challenges

- **The Hydrologic Simulation Program–Python (HSP^2) Conversion Project**
  - Project Goals
  - HSP^2 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

UNITED STATES • ENVIRONMENTAL PROTECTION AGENCY

USGS science for a changing world

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**

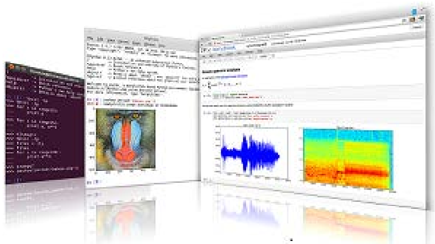
Project Goal and Objectives

- **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.**

RESPEC

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
 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

IPython
Enhanced Interactive Console

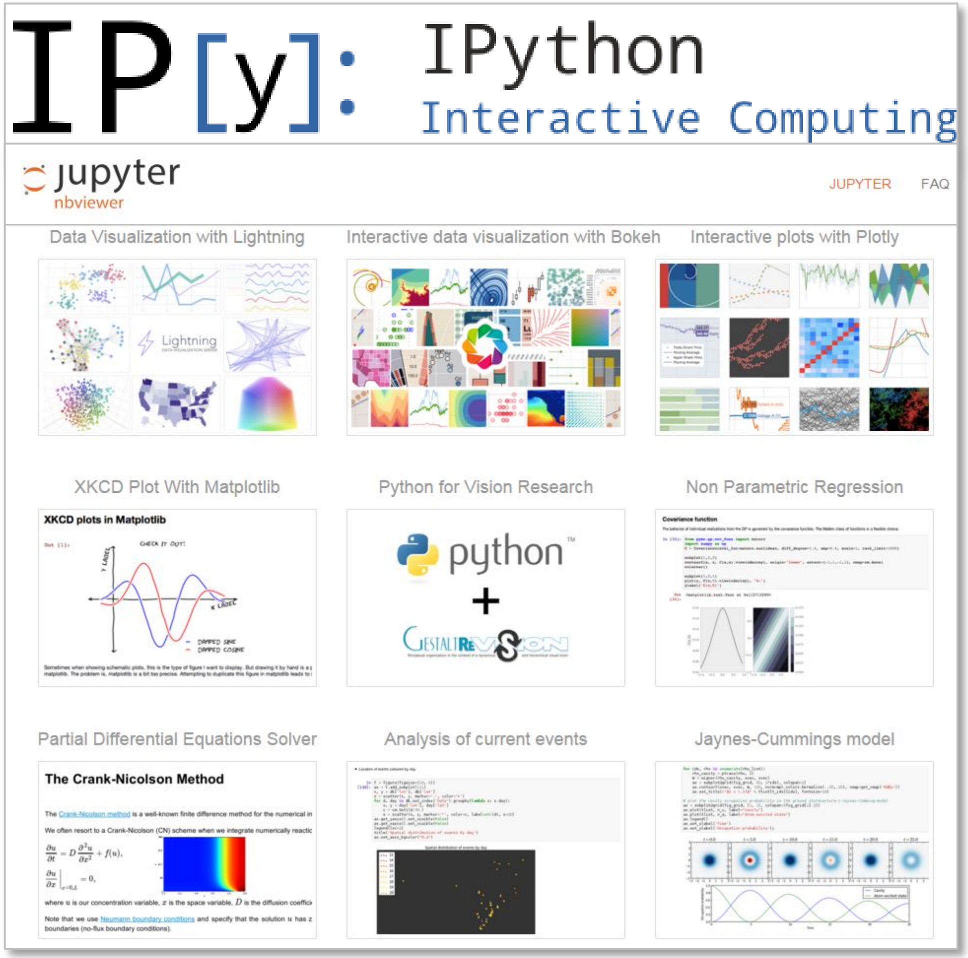
Sympy
Symbolic mathematics

pandas
Data structures & analysis

Why Python?

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

- **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



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.**

HSP^2 Modeling System

HSP^2 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 HSP^2**
  - HSP^2 has line numbers of corresponding HSPF code as comments
  - Module names are also traced between HSPF and HSP^2 equivalent code
  - Variable names are basically unchanged for traceability
  - Many comments from the HSPF code have been carried to the HSP^2 code
- **Verification Tools**
  - Rapidly compare the results of HSPF verification tests with HSP^2 results
  - HDFView to check results of data storage operations
- **Version Control**
  - GitHub

HSP^2 Modeling System

Lines of Code
Typically 5 – 10 times less

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

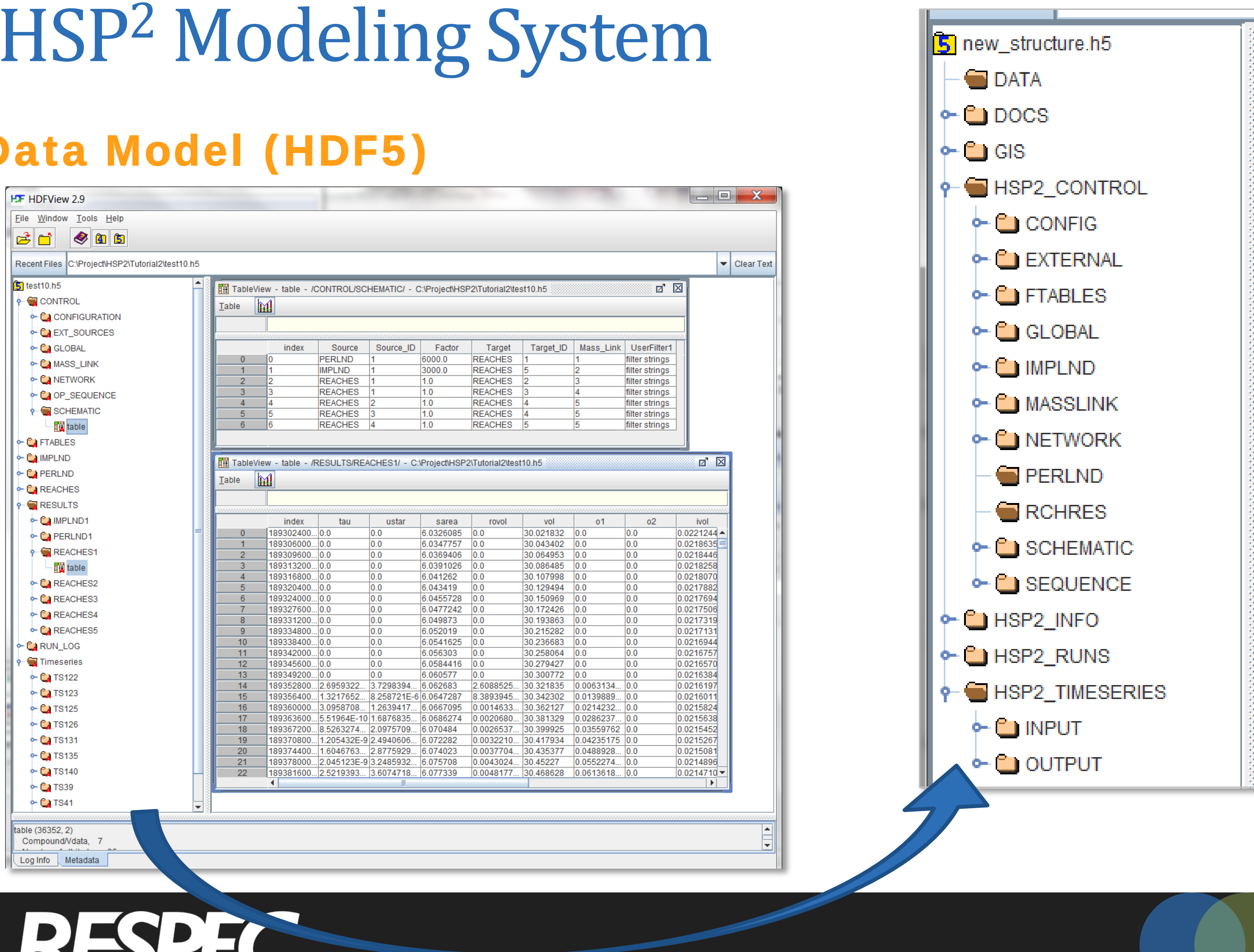
Code Example

```
if CSNOFG:                                                            #$974
    """ snow is being considered - allow for it find the moisture supplied
    to interception storage.  rainf is rainfall in inches/ivl. adjust for
    fraction of land segment covered by snow. wyield is the water yielded by
    the snowpack in inches/ivl. it has already been adjusted to an effective
    yield over the entire land segment """

    airtmpV = typeV(ts['AIRTMP'])                                     #$983-995
    rainfV  = typeV(ts['rainf'])                                      #$1005
    snocovV = typeV(ts['snocov'])                                     #$1010
    wyieldV = typeV(ts['wyield'])                                     #$1015

    supyV = rainfV * (1.0 - snocovV) + wyieldV                        #$1030

    petadjV = (1.0 - forest) * (1.0 - snocovV) + forest               #$1037
    petadjV[(airtmpV < petmax) & (petadjV > 0.5)] = 0.5               #$1039,1045-1048
    petadjV[airtmpV < petmin] = 0.0                                   #$1041-1044
    ts['petadj'] = petadjV
```

Fortran Line #s

HSP^2 Modeling System

Data Model (HDF5)

- 

HSP^2 Modeling System

Library of Tutorials Developed to Demonstrate Functionality!

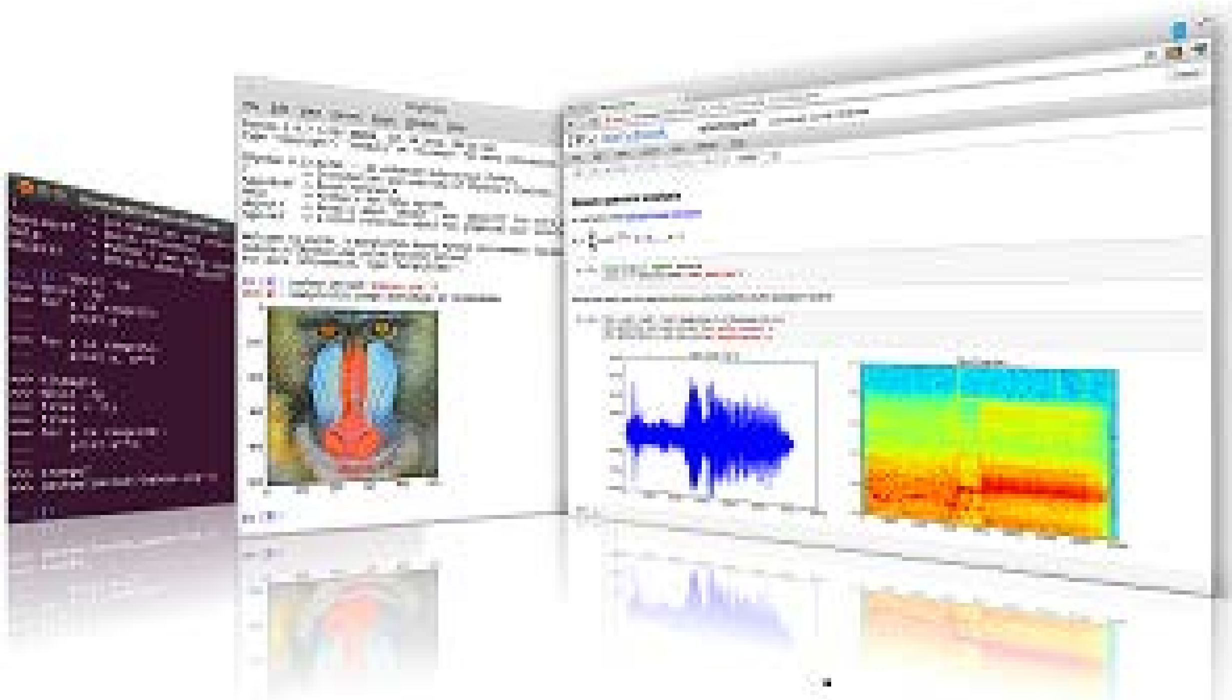
Summary of New Features

- **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)**
  - HSP^2 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