Intrusive and Non-intrusive Uncertainty Propagation

UQTk Example

Description of input parameters

- Command-line usage:

```
./prob2.py <pctype> <pcord> <method1> [<method2>]  [<method3>]
```

- Performs parametric uncertainty propagation using three methods
  - *NISP*: Non-intrusive spectral projection using quadrature integration
  - *ISP*: Intrusive spectral projection
  - *NISP_MC*: Non-intrusive spectral projection using Monte-Carlo integration
- Requires xml input template file *prob2_surf_rxn.in.xml.templ*
  - Default setting is: param_b is uncertain Gaussian with stdev = 10% of the mean
- Parameters defined at the beginning of the file:
  - *pctype*: The type of PC, supports 'HG', 'LU', 'GLG', 'JB'
  - *pcord*: The order of output PC expansion
  - *methodX*: NISP ISP or NISP_MC
  - *nsam*: Number of samples requested for NISP Monte-Carlo [Hardwired]

Steps for Non-intrusive Spectral Projection

SurfRxnNISP.x (.cpp) and SurfRxnNISP_MC.x (.cpp)

$$f(\xi) = \sum_k c_k \Psi_k(\xi) \qquad c_k = \frac{\langle f(\xi)\Psi_k(\xi)\rangle}{\langle \Psi_k^2(\xi)\rangle}$$

$$\langle f(\xi)\Psi_k(\xi)\rangle = \int f(\xi)\Psi_k(\xi)\pi(\xi)d\xi \approx \underbrace{\left[\sum_q f(\xi_q)\Psi_k(\xi_q)w_q\right]}_{NISP} \text{ or } \underbrace{\left[\frac{1}{N}\sum_s f(\xi_s)\Psi_k(\xi_s)\right]}_{NISP\_MC}$$

1. Read XML file
2. Create a PC object with or without quadrature
   - NISP: `PCSet myPCSet("NISP",order,dim,pcType,0.0,1.0)`
   - NISP_MC: `PCSet myPCSet("NISPnoq",order,dim,pcType,0.0,1.0)`
3. Get the quadrature points or generate Monte-Carlo samples
   - NISP: `myPCSet.GetQuadPoints(qdpts)`
   - NISP_MC: `myPCSet.DrawSampleVar(samPts)`
4. Create input PC objects and evaluate input parameters corresponding to quadrature points
5. Step forward in time
   - Collect values for all input parameter samples
   - Perform Galerkin projection or Monte-Carlo integration
   - Write the PC modes and derived first two moments to files

Steps for Intrusive Spectral Projection

SurfRxnISP.x (.cpp)

1. Read XML file
2. Create a PC object for intrusive propagation

   ```
   PCSet myPCSet("ISP",order,dim,pcType,0.0,1.0)
   ```
3. Represent state variables and all parameters with their PC coefficients
   - $u \rightarrow \{u_k\}$, $v \rightarrow \{v_k\}$, $w \rightarrow \{w_k\}$, $z \rightarrow \{z_k\}$,
   - $a \rightarrow \{a_k\}$, $b \rightarrow \{b_k\}$, $c \rightarrow \{c_k\}$, $d \rightarrow \{d_k\}$, $e \rightarrow \{e_k\}$, $f \rightarrow \{f_k\}$.
4. Step forward in time according to PC arithmetics, *e.g.*

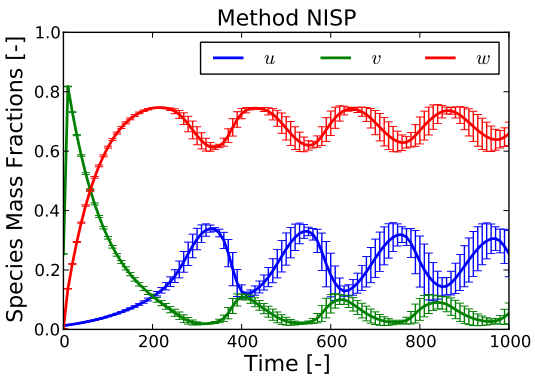
$a \cdot u \rightarrow \{(a \cdot u)_k\}$ with

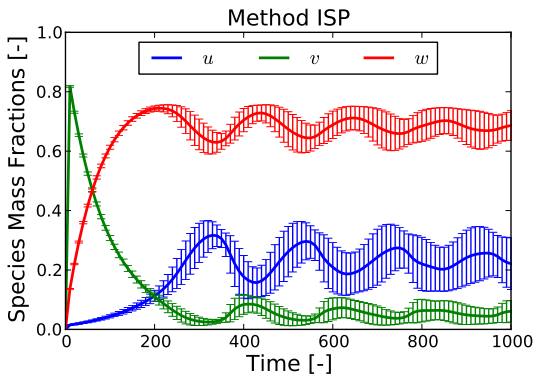
$$a \cdot u = \left(\sum_i a_i \Psi_i(\xi) \right) \left(\sum_j u_j \Psi_j(\xi) \right) = \sum_k \underbrace{\left(\sum_{i,j} a_i u_j \frac{\langle \Psi_i \Psi_j \Psi_k \rangle}{\langle \Psi_k^2 \rangle} \right)}_{(a \cdot u)_k} \Psi_k(\xi)$$

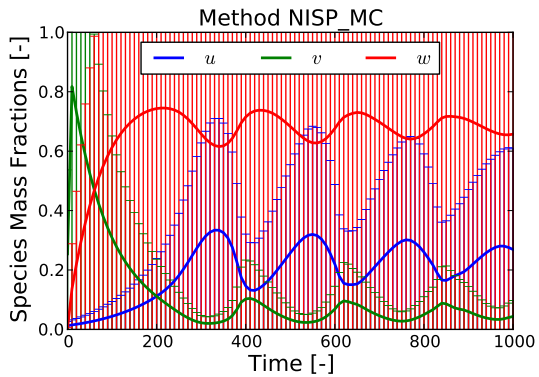
Plotting utilities - time series

```
./plSurfRxnMstd.py NISP
./plSurfRxnMstd.py ISP
./plSurfRxnMstd.py NISP_MC
```

Plot time evolution of mean and standard deviations of all three species with all three methods







Plotting utilities - PDFs

`./plPDF_method.py <species>  <qoi> <pctype> <pcord> <method1> [<method2>] [<method3>]`

e.g. `./plPDF_method.py u ave HG 3 NISP ISP`

Sample the PC representation and plot the PDFs of time-average (ave) or the final time value (tf) for all three species.