Surface Reaction Model Uncertainty Quantification

Intrusive and Non-Intrusive Methods

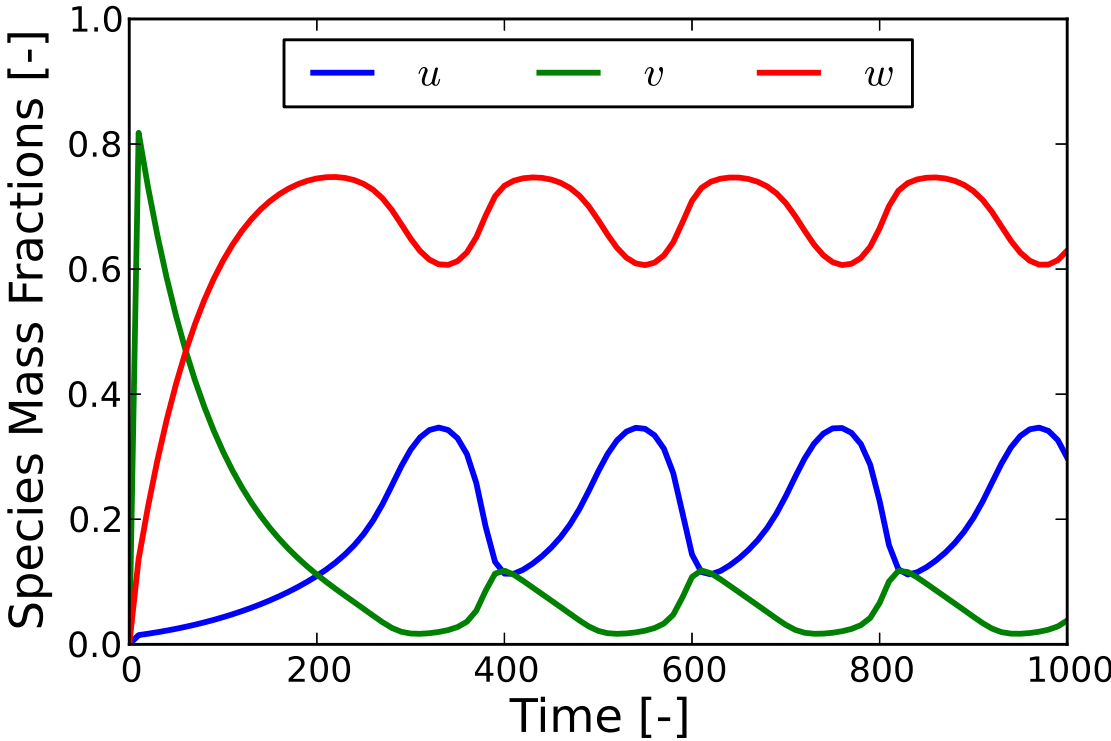
UQTk Example

Surface Reaction Model

3 ODEs for a monomer (u), dimer (v), and inert species (w) adsorbing onto a surface out of gas phase.

$$\begin{aligned}
\frac{\mathrm{d}u}{\mathrm{d}t} &= az - cu - 4duv \\
\frac{\mathrm{d}v}{\mathrm{d}t} &= 2bz^2 - 4duv \\
\frac{\mathrm{d}w}{\mathrm{d}t} &= ez - fw \\
z &= 1 - u - v - w
\end{aligned}$$

$$u(0) = v(0) = w(0) = 0.0$$



Oscillatory behavior for $b \in [20.2, 21.2]$

$a = 1.6 \quad b = 20.75 \quad c = 0.04 \quad d = 1.0 \quad e = 0.36 \quad f = 0.016$

(Vigil *et al.*, Phys. Rev. E., 1996; Makeev *et al.*, J. Chem. Phys., 2002)

Surface Reaction Model: Intrusive Spectral Propagation (ISP) of Uncertainty

- Assume PCE for uncertain parameter b and for the output variables, u, v, w
- Substitute PCEs into the governing equations
- Project the governing equations onto the PC basis functions
  - Multiply with Ψ_k and take the expectation
- Apply pseudo-spectral approximations where necessary
- UQTk elementary operations

Surface Reaction Model: Specify PCEs for inputs and outputs

Represent uncertain inputs with PCEs with known coefficients:

$$b = \sum_{i=0}^{P} b_i \Psi_i(\xi)$$

Represent all uncertain variables with PCEs with unknown coefficients:

$$u = \sum_{i=0}^{P} u_i \Psi_i(\xi) \quad v = \sum_{i=0}^{P} v_i \Psi_i(\xi) \quad w = \sum_{i=0}^{P} w_i \Psi_i(\xi) \quad z = \sum_{i=0}^{P} z_i \Psi_i(\xi)$$

Surface Reaction Model: Substitute PCEs into governing equations and project onto basis functions

$$
\begin{aligned}
\frac{\mathrm{d}u}{\mathrm{d}t} &= az - cu - 4duv \\
\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} u_i\Psi_i &= a\sum_{i=0}^{P} z_i\Psi_i - c\sum_{i=0}^{P} u_i\Psi_i - 4d\sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \\
\left\langle \Psi_k \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} u_i\Psi_i \right\rangle &= \left\langle a\Psi_k \sum_{i=0}^{P} z_i\Psi_i \right\rangle - \left\langle c\Psi_k \sum_{i=0}^{P} u_i\Psi_i \right\rangle \\
&- \left\langle 4d\Psi_k \sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \right\rangle
\end{aligned}
$$

Surface Reaction Model: Reorganize terms

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} u_k \left\langle \Psi_k^2 \right\rangle &= a z_k \left\langle \Psi_k^2 \right\rangle - c u_k \left\langle \Psi_k^2 \right\rangle - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \left\langle \Psi_i \Psi_j \Psi_k \right\rangle \\
\frac{\mathrm{d}}{\mathrm{d}t} u_k &= a z_k - c u_k - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle} \\
\frac{\mathrm{d}}{\mathrm{d}t} u_k &= a z_k - c u_k - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j C_{ijk}
\end{aligned}$$

- Triple products $C_{ijk} = \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle}$ can be pre-computed and stored for repeated use

Surface Reaction Model: Substitute PCEs into governing equations and project onto basis functions

$$
\begin{aligned}
\frac{\mathrm{d}v}{\mathrm{d}t} &= 2bz^2 - 4duv \\
\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} v_i\Psi_i &= 2\sum_{h=0}^{P} b_h\Psi_h \sum_{i=0}^{P} z_i\Psi_i \sum_{j=0}^{P} z_j\Psi_j - 4d\sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \\
\left\langle \Psi_k \frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=0}^{P} v_i\Psi_i \right\rangle &= \left\langle 2\Psi_k \sum_{h=0}^{P} b_h\Psi_h \sum_{i=0}^{P} z_i\Psi_i \sum_{j=0}^{P} z_j\Psi_j \right\rangle \\
&- \left\langle 4d\Psi_k \sum_{i=0}^{P} u_i\Psi_i \sum_{j=0}^{P} v_j\Psi_j \right\rangle
\end{aligned}
$$

Surface Reaction Model: Reorganize terms

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k \left\langle \Psi_k^2 \right\rangle = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j \left\langle \Psi_h \Psi_i \Psi_j \Psi_k \right\rangle - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \left\langle \Psi_i \Psi_j \Psi_k \right\rangle$$

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j \frac{\left\langle \Psi_h \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle} - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j \frac{\left\langle \Psi_i \Psi_j \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} v_k = 2 \sum_{h=0}^{P} \sum_{i=0}^{P} \sum_{j=0}^{P} b_h z_i z_j D_{hijk} - 4d \sum_{i=0}^{P} \sum_{j=0}^{P} u_i v_j C_{ijk}$$

- Pre-computing and storing the quad product D_{hijk} becomes cumbersome
- Use pseudo-spectral approach instead

Surface Reaction Model: Pseudo-Spectral approach for products

- Introduce auxiliary variable $g = z^2$

$$\begin{aligned} g &= z^2 \\ f = 2bz^2 &= 2bg \end{aligned}$$

$$\begin{aligned} g_k &= \sum_{i=0}^{P}\sum_{j=0}^{P} z_i z_j C_{ijk} \\ f_k &= 2\sum_{i=0}^{P}\sum_{j=0}^{P} b_i g_j C_{ijk} \end{aligned}$$

- Limits the complexity of computing product terms
  - Higher products can be computed by repeated use of the same binary product rule
- Does introduce errors if order of PCE is not large enough

Surface Reaction Model: UQTk implementation

```
// Build du/dt = a*z - c*u - 4.0*d*u*v
aPCSet.Multiply(z,a,dummy1);            // dummy1 = a*z
aPCSet.Multiply(u,c,dummy2);            // dummy2 = c*u
aPCSet.SubtractInPlace(dummy1,dummy2); // dummy1 = a*z - c*u
aPCSet.Prod(u,v,dummy2);                // dummy2 = u*v
aPCSet.MultiplyInPlace(dummy2,4.e0*d); // dummy2 = 4.0*d*u*v
aPCSet.Subtract(dummy1,dummy2,dudt);    // dudt = a*z - c*u - 4.0*d*u*v
```

- All operations are replaced with their equivalent intrusive UQ counterparts
- Results in a set of coupled ODEs for the PC coefficients
  - u, v, w, z represent vector of PC coefficients
- This set of equations is integrated to get the evolution of the PC coefficients in time

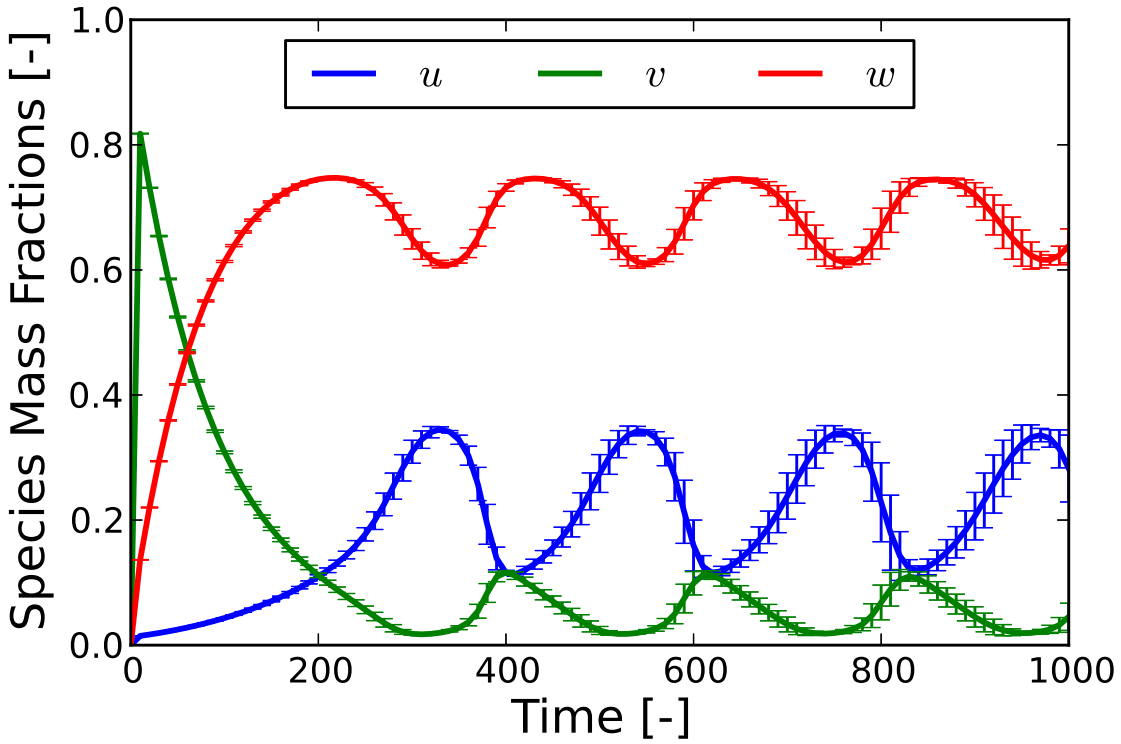
Surface Reaction Model: Second equation implementation

```
// Build dv/dt = 2.0*b*z*z - 4.0*d*u*v
aPCSet.Prod(z,z,dummy1);                 // dummy1 = z*z
aPCSet.Prod(dummy1,b,dummy2);            // dummy2 = b*z*z
aPCSet.Multiply(dummy2,2.e0,dummy1);     // dummy1 = 2.0*b*z*z
aPCSet.Prod(u,v,dummy2);                 // dummy2 = u*v
aPCSet.MultiplyInPlace(dummy2,4.e0*d); // dummy2 = 4.0*d*u*v
aPCSet.Subtract(dummy1,dummy2,dvdt);     // dvdt = 2.0*b*z*z - 4.0*d*u*v

// Build dw/dt = e*z - f*w
aPCSet.Multiply(z,e,dummy1);             // dummy1 = e*z
aPCSet.Multiply(w,f,dummy2);             // dummy2 = f*w
aPCSet.Subtract(dummy1,dummy2,dwdt);     // dwdt = e*z - f*w
```

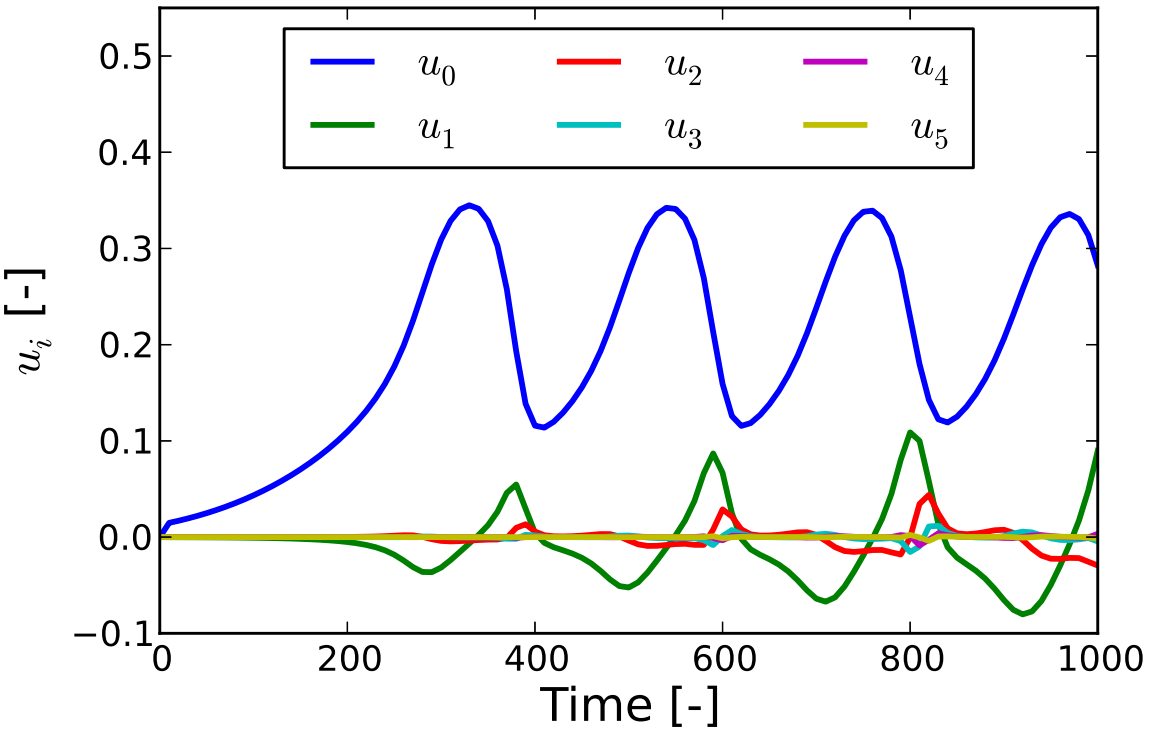
- Dummy variables used where needed to build the terms in the equations

Surface Reaction Model: ISP results



- Assume 0.5% uncertainty in *b* around nominal value
- Legendre-Uniform intrusive PC
- Mean and standard deviation for *u*, *v*, and *w*
- Uncertainty grows in time

Surface Reaction Model: ISP results



- Modes of u
- Modes decay with higher order
- Amplitudes of oscillations of higher order modes grow in time

Non-Intrusive propagation of uncertainty - Projection

Galerkin Projection

$$u_k = \frac{\langle u\Psi_k \rangle}{\langle \Psi_k^2 \rangle} = \frac{1}{\langle \Psi_k^2 \rangle} \int u\Psi_k(\xi) w(\xi) d\xi, \quad k = 0, \ldots, P$$

Evaluate projection integrals numerically

- Pick samples of uncertain parameters, e.g. $b(\xi)$ by sampling the germ ξ
- Run deterministic forward model for each of the sampled input parameter values $b_i = b(\xi_i)$
- Integration depends on sampling approach
  - Random Sampling: $\langle u\Psi_k \rangle = \frac{1}{N_s} \sum_{i=1}^{N_s} u(b_i)\Psi_k(\xi_i)$
  - Quadrature: $\langle u\Psi_k \rangle = \sum_{i=1}^{N_q} q_i u(b_i)\Psi_k(\xi_i)$

Reconstruct uncertain model output

$$u(x, t; \theta) = \sum_{k=0}^{P} u_k(x, t)\Psi_k(\xi(\theta))$$

Random Sampling approaches

- Evaluate integral through sampling

$$\int u\Psi_k(\xi)w(\xi)d\xi = \frac{1}{N_s}\sum_{i=1}^{N_s} u(\xi_i)\Psi_k(\xi_i)$$

- Samples are drawn according to the distribution of ξ
  - Monte-Carlo (MC)
  - Latin-Hypercube-Sampling (LHS)
- Pros:
  - Can be easily made fault tolerant
  - Sometimes random samples is all we have
- Cons: slow convergence, but less dependent on number of stochastic dimensions

Quadrature approaches

- Numerically evaluate integrals in Galerkin projection

$$\int u\Psi_k(\xi)w(\xi)d\xi = \sum_{i=1}^{N_q} q_i u(\xi_i)\Psi_k(\xi_i)$$

- Gauss quadrature rules are very efficient
  - ξ are quadrature points, with corresponding weights q_i
  - N_q quadrature points can integrate polynomial of order $2N_q - 1$ exactly
  - Gauss-Hermite and Gauss-Legendre quadrature tailored to specific choices of the weight function $w(\xi)$
  - As a rule of thumb, $p + 1$ quadrature points are needed for Galerkin projection of PCE of order p
    - If both u and Ψ_k are of order p, then integrand is of order $2p$
    - $2p \leq 2N_q - 1$ or $N_q \geq p + \frac{1}{2}$
    - Only exact if u is indeed a polynomial of order $\leq p$
- Pros:
  - Can use existing codes as black box to evaluate $u(\xi_i)$
  - Embarrassingly parallel
- Cons: Tensor product rule for d dimensions requires N_q^d samples

Non-Intrusive propagation of uncertainty - Collocation

- Collocation techniques minimize errors at sample points
  - $\sum_{k=0}^{P} u_k \Psi_k(\xi_i) = u(\xi_i) \ , \quad i = 1, \ldots, N_c$
  - Can use interpolation, e.g. Lagrange interpolants
  - Or use regression approaches: $P + 1$ degrees of freedom to fit N_c points
- Pros: can position points where most accuracy desired
- Cons:

Surface Reaction Model: NISP implementation in UQTk

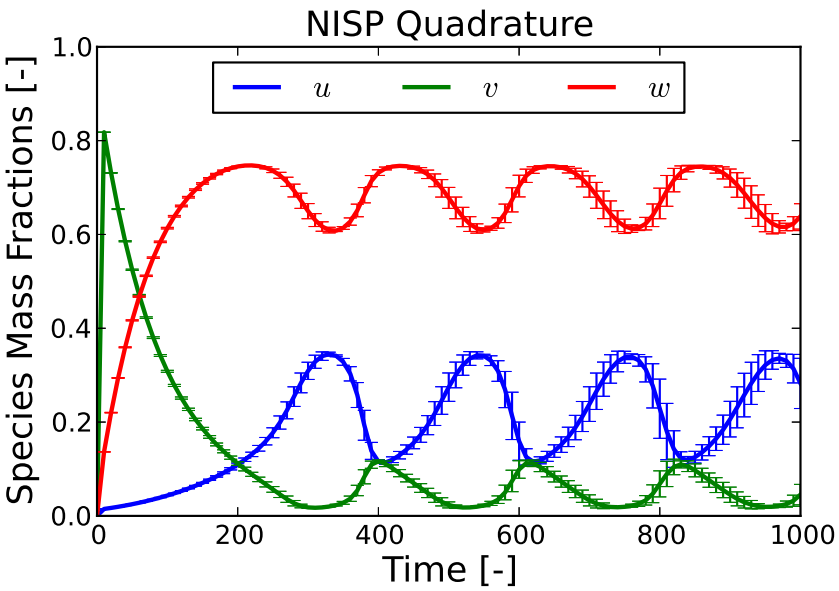
Quadrature:

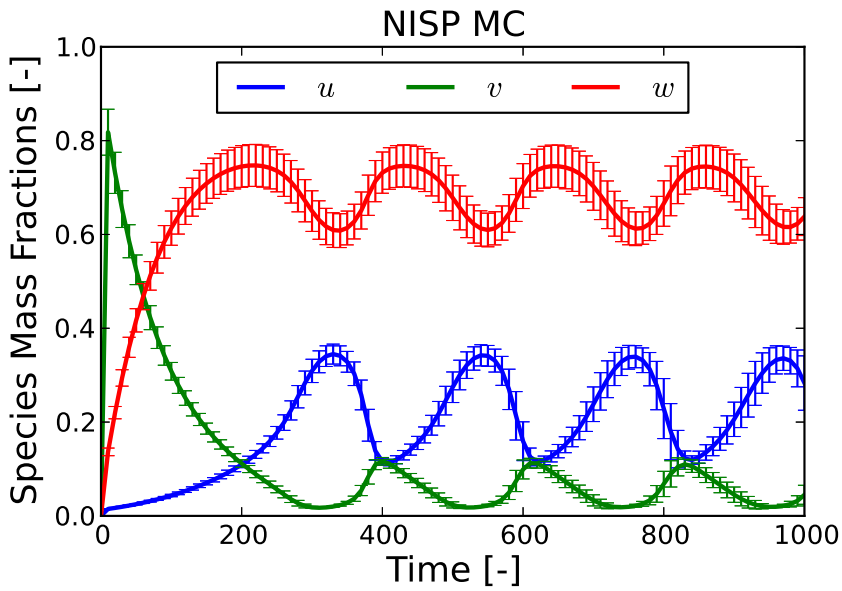
```
// Get the quadrature points
int nQdpts=myPCSet.GetNQuadPoints();
double* qdpts=new double[nQdpts];
myPCSet.GetQuadPoints(qdpts);
...
// Evaluate parameter at quad pts
for(int i=0;i<nQdpts;i++){
  bval[i]=myPCSet.EvalPC(b,&qdpts[i]);
}
...
// Run model for all samples
for(int i=0;i<nQdpts;i++){
  u_val[i] = ...
}
// Spectral projection
myPCSet.GalerkProjection(u_val,u);
myPCSet.GalerkProjection(v_val,v);
myPCSet.GalerkProjection(w_val,w);
```

Monte-Carlo Sampling:

```
// Get the sample points
int nSamples=1000;
Array2D<double> samPts(nSamples,dim);
myPCSet.DrawSampleVar(samPts);
...
// Evaluate parameter at sample pts
for(int i=0;i<nSamples;i++){
  ... // select samPt from samPts
  bval[i]=myPCSet.EvalPC(b,&samPt)
}
...
// Run model for all samples
for(int i=0;i<nSamples;i++){
  u_val[i] = ...
}
// Spectral projection
myPCSet.GalerkProjectionMC(samPts,u_val,u);
myPCSet.GalerkProjectionMC(samPts,v_val,v);
myPCSet.GalerkProjectionMC(samPts,w_val,w);
```

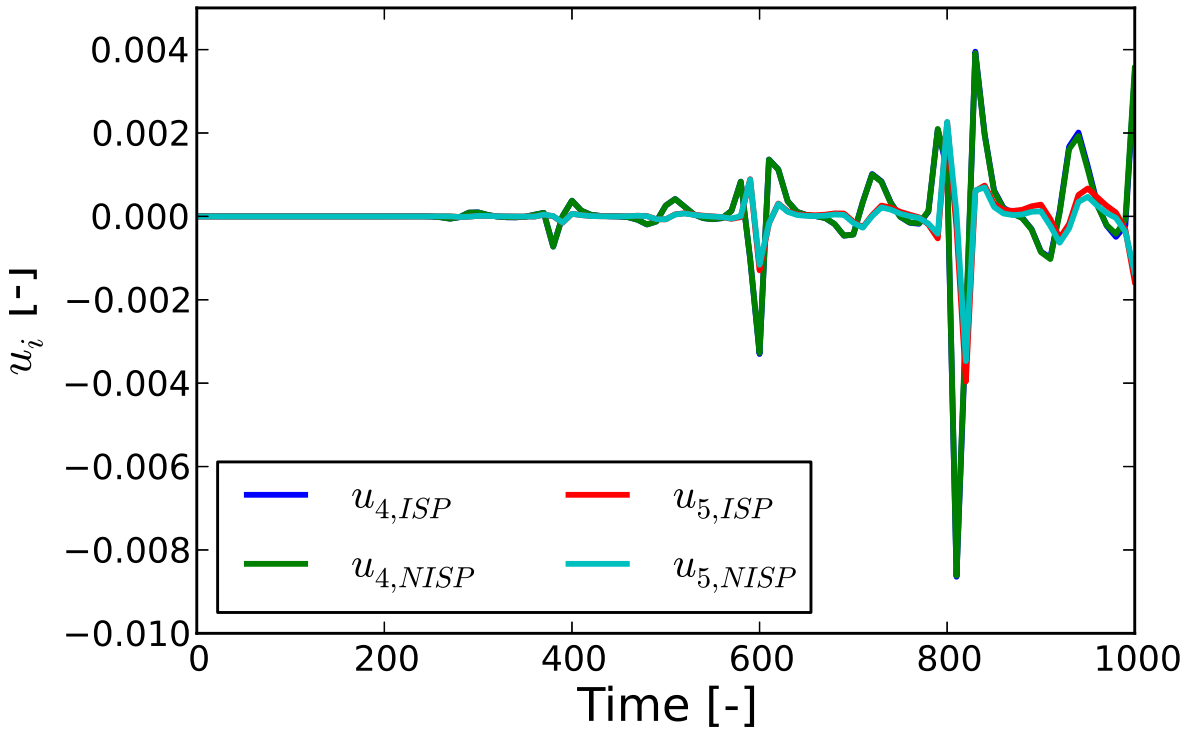
Surface Reaction Model: NISP results





- Mean and standard deviation for u, v, and w
- Quadrature approach agrees well with ISP approach using 6 quadrature points
- Monte Carlo sampling approach converges slowly
  - With a 1000 samples, results are quite different from ISP and NISP

Surface Reaction Model: Comparison ISP and NISP



- Lower order modes agree perfectly
- Very small differences in higher order modes
  - Difference increases with time

Further Reading

- R. Vigil and F. Willmore, “Oscillatory dynamics in a heterogeneous surface reaction: Breakdown of the mean-field approximation.,” *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, vol. 54, no. 2, pp. 1225-1231, Aug. 1996.
- A. G. Makeev, D. Maroudas, and I. G. Kevrekidis, “ ’Coarse’ stability and bifurcation analysis using stochastic simulators: Kinetic Monte Carlo examples,” *J. Chem. Phys.*, vol. 116, no. 23, p. 10083, 2002.
- B. J. Debusschere, H. N. Najm, A. Matta, O. M. Knio, R. G. Ghanem, and O. P. Le Maître, ”Protein labeling reactions in electrochemical microchannel flow: Numerical simulation and uncertainty propagation,” *Phys. Fluids*, vol. 15, no. 8, p. 2238, 2003.
- B. Debusschere, H. Najm, P. Pébay, O. Knio, R. Ghanem and O. Le Maître, “Numerical Challenges in the Use of Polynomial Chaos Representations for Stochastic Processes", *SIAM J. Sci. Comp.*, 26:2, 2004.
- O. P. Le Maître and O. M. Knio, “Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics” (Scientific Computation), 1st ed. Springer, 2010.