MECA0010 – Reliability and stochastic modeling of engineered systems

Uncertainty quantification

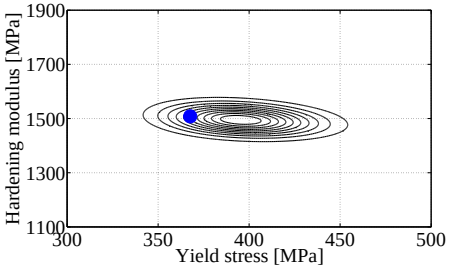
Part 4 of 4 — Stochastic expansion methods and surrogate modeling

Maarten Arnst

October 25, 2017

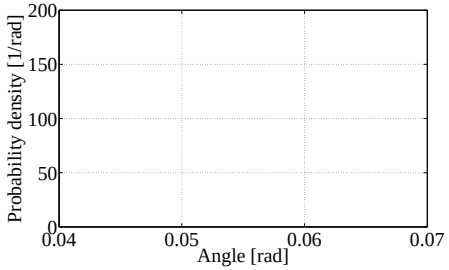
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



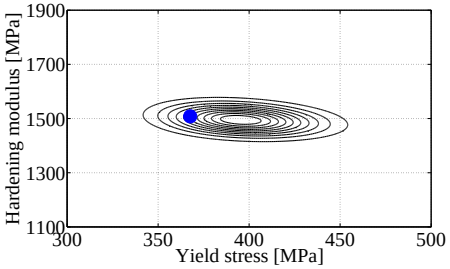
→

→



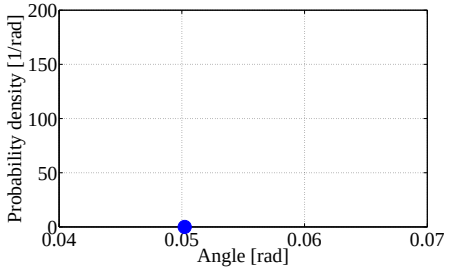
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



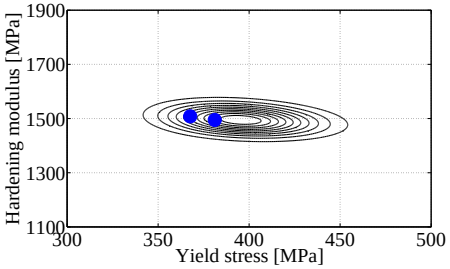
→

→



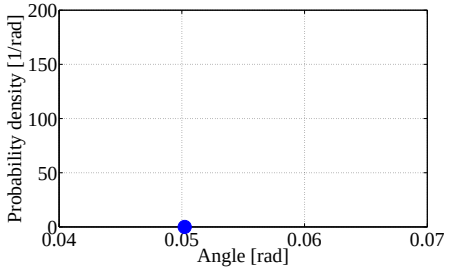
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



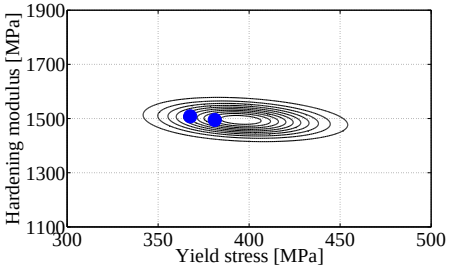
→

→



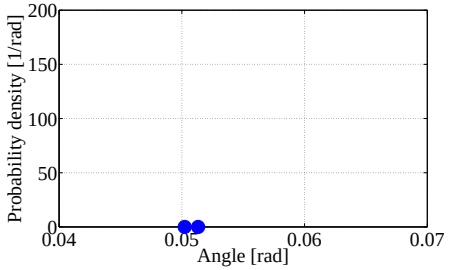
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



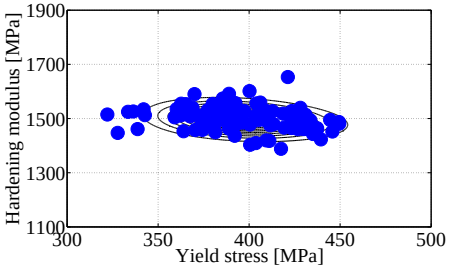
→

→



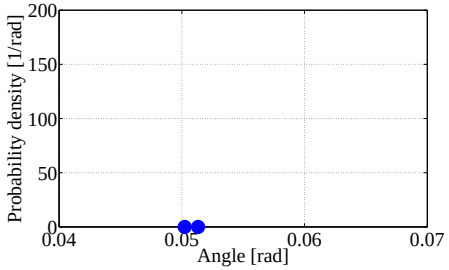
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



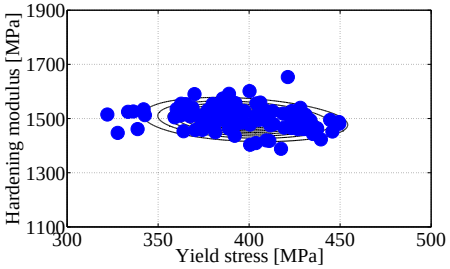
→

→



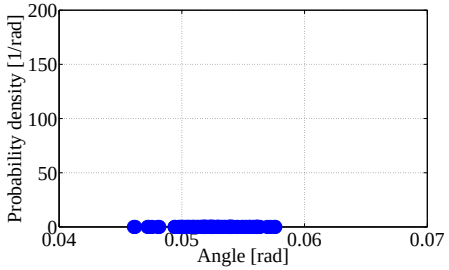
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



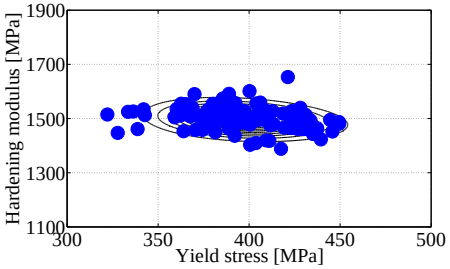
→

→



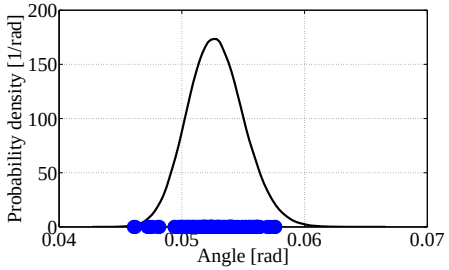
Manufacturing tolerances in metal forming

- **Monte Carlo method**:



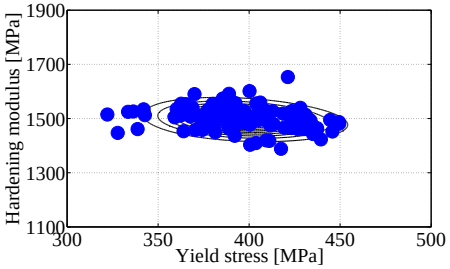
→

→



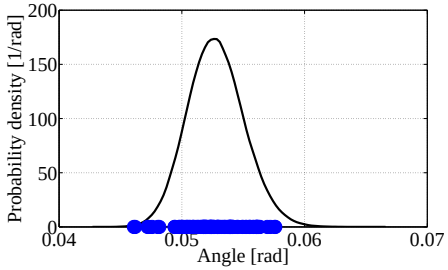
Manufacturing tolerances in metal forming

- **Monte Carlo method**:

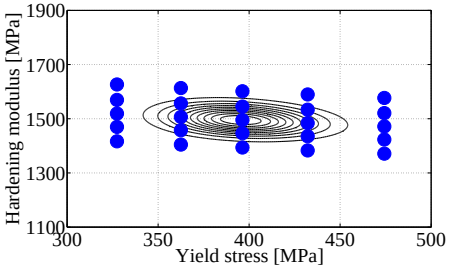


→

→

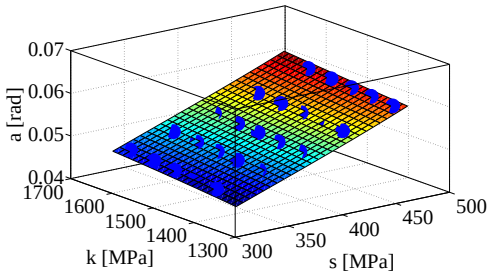


- **Stochastic expansion method**:



→

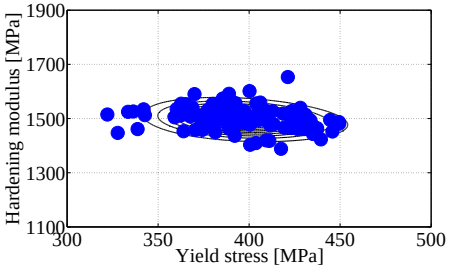
↓



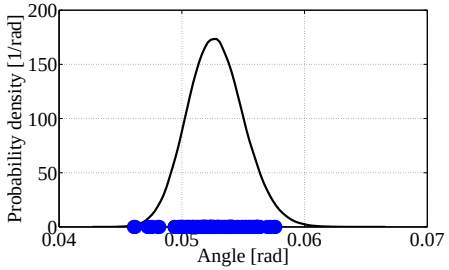
Computationally inexpensive surrogate model

Manufacturing tolerances in metal forming

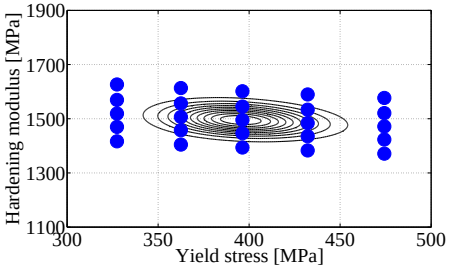
- **Monte Carlo method**:



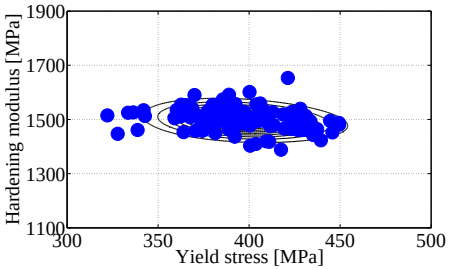
→ Metafor (nonlinear Finite Element Software, LTAS-MN²L) →



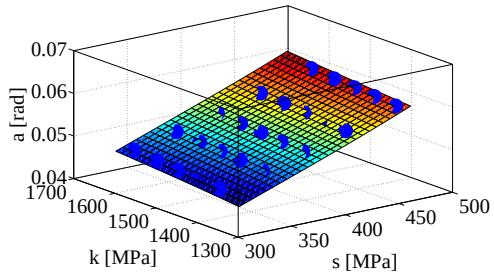
- **Stochastic expansion method**:



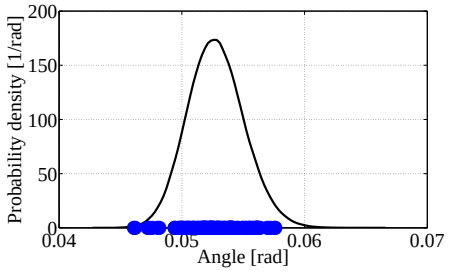
→ Metafor (nonlinear Finite Element Software, LTAS-MN²L) ↓



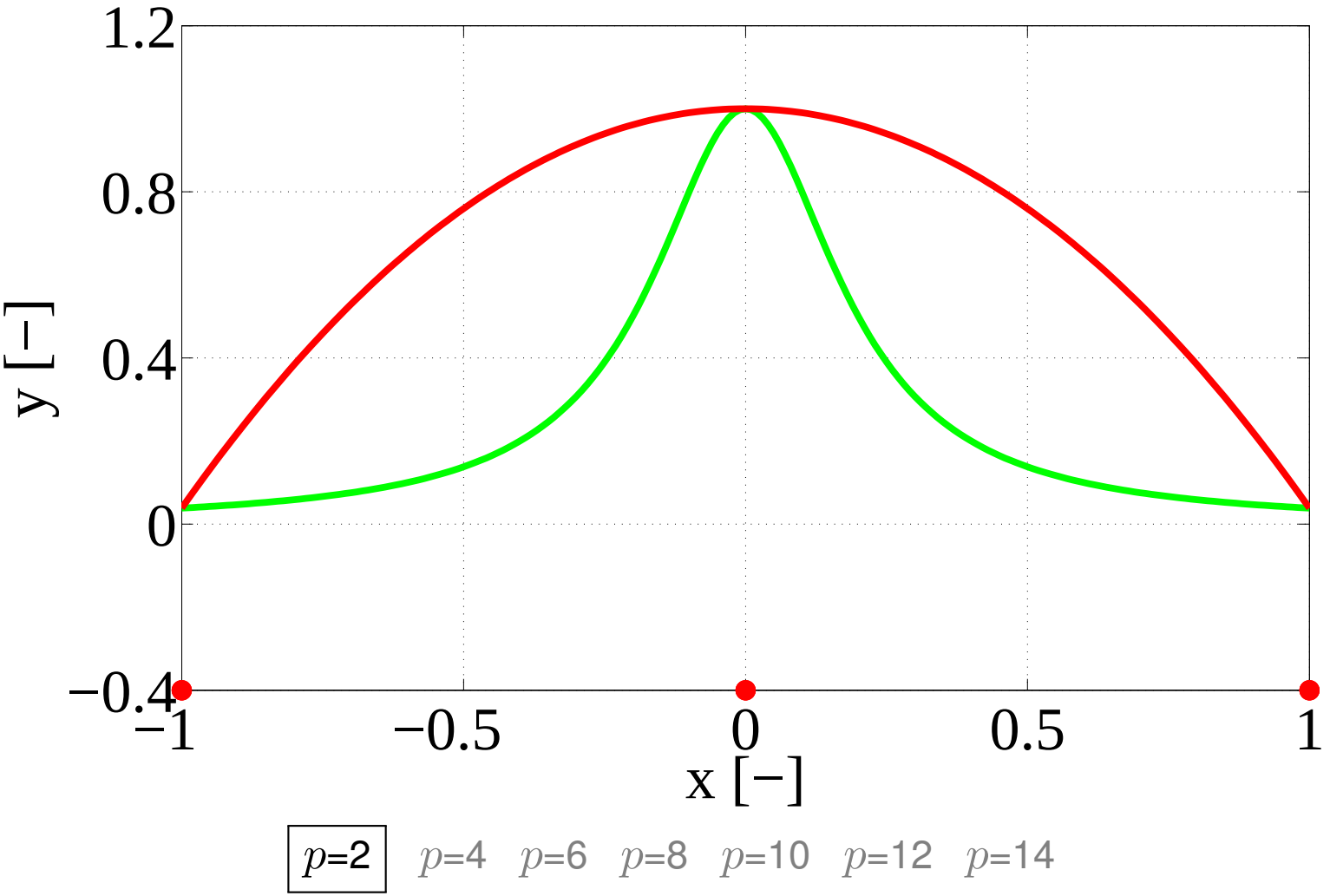
→



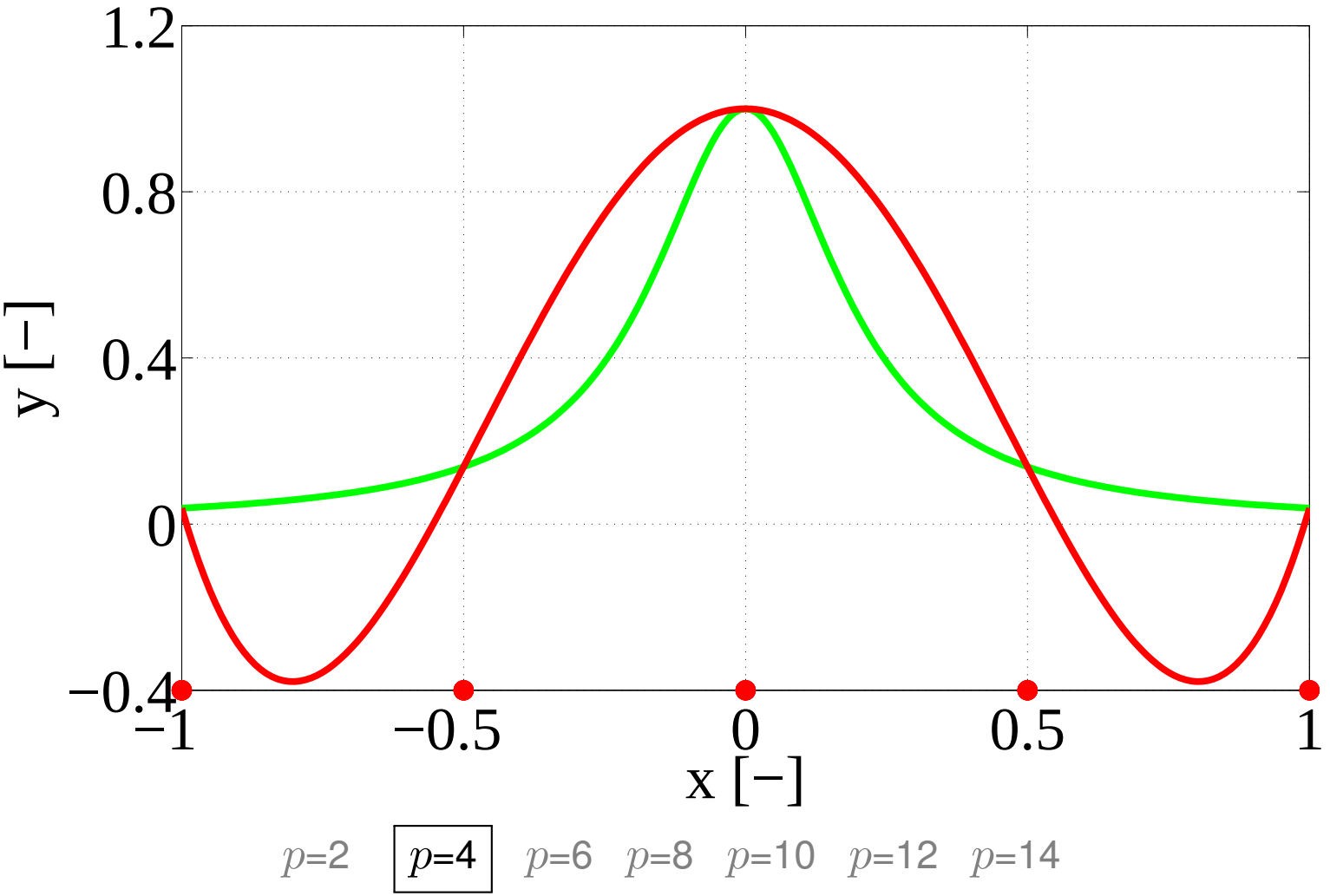
→



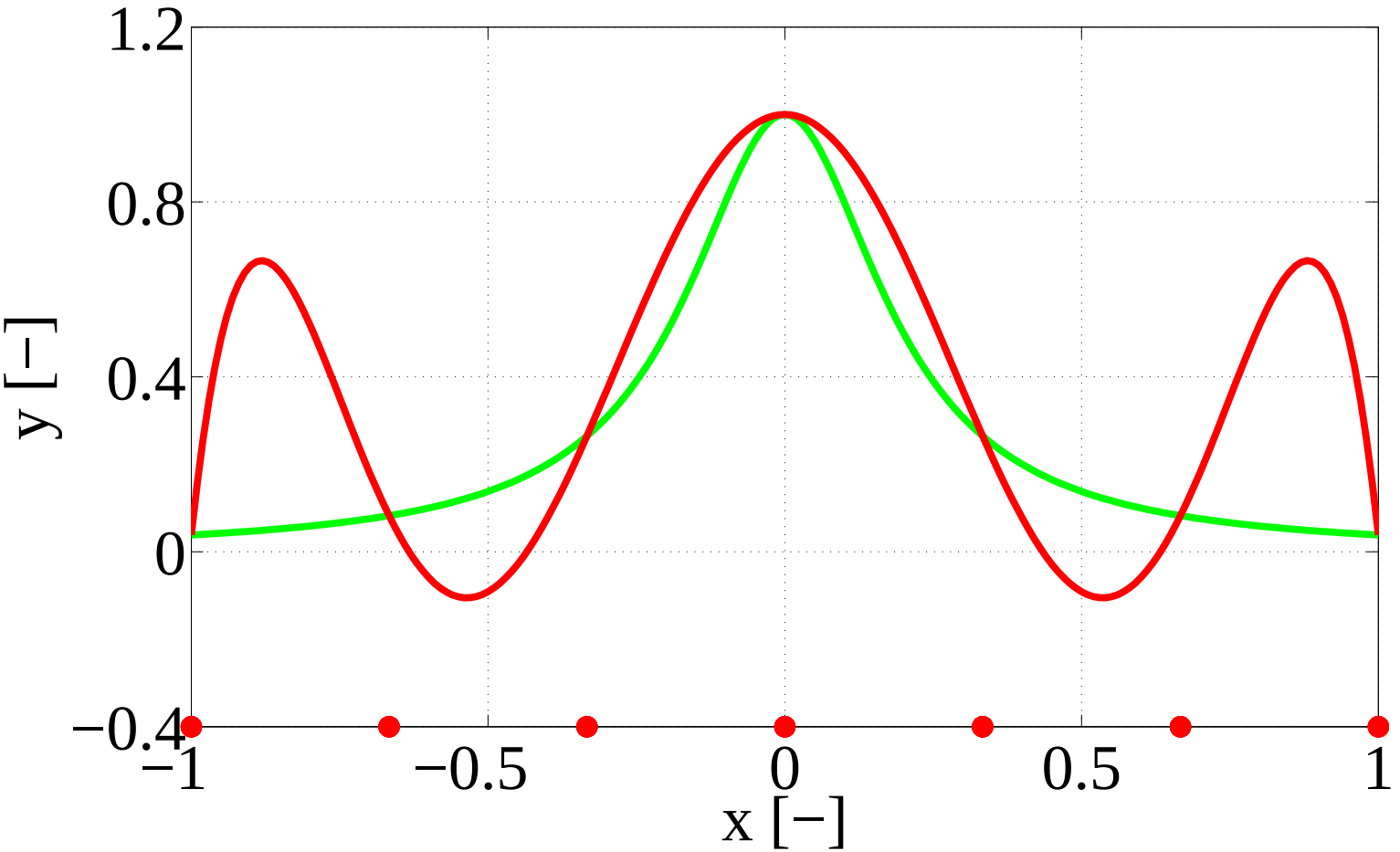
Computationally inexpensive surrogate model



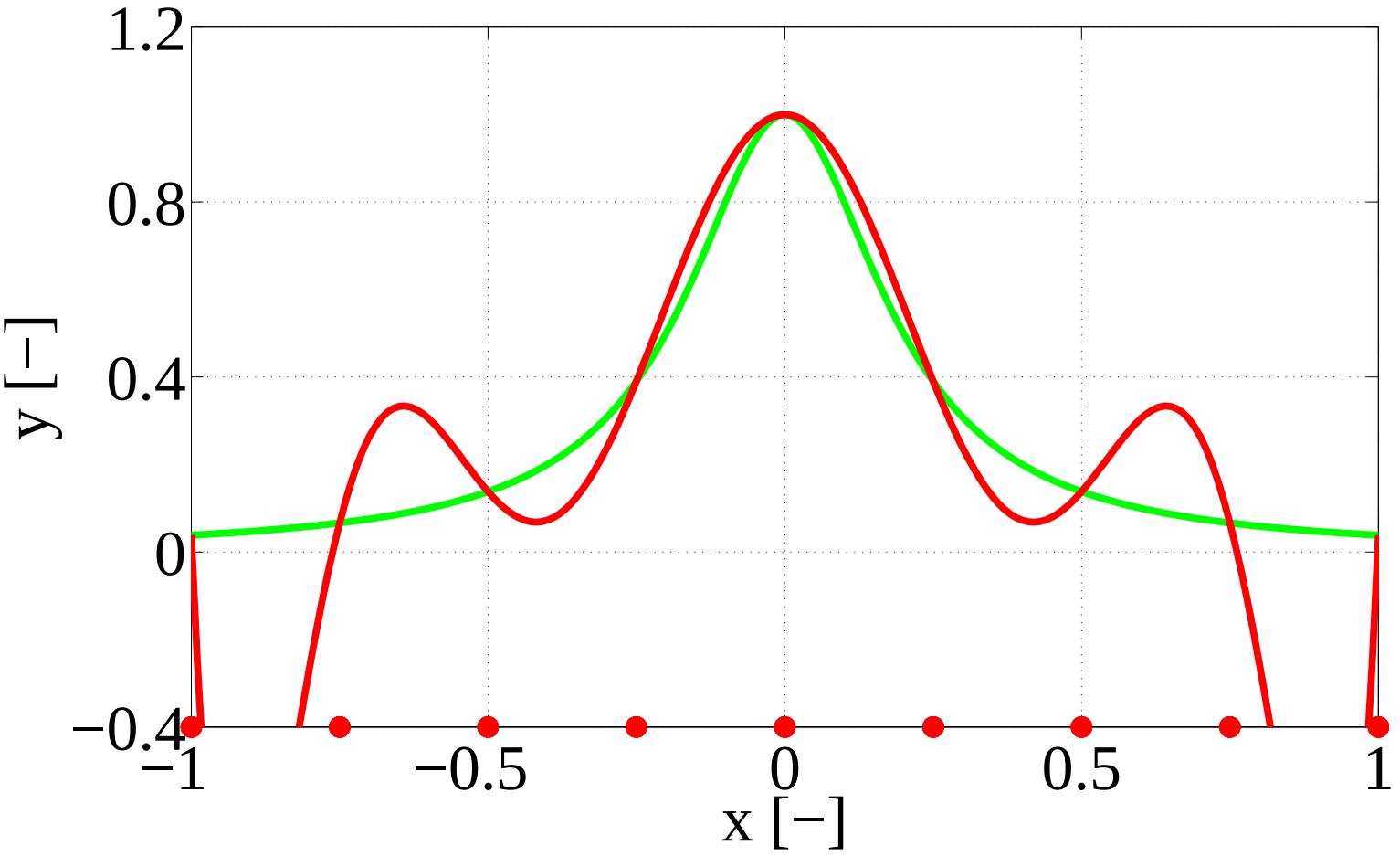
It's not so easy! Interpolation with equidistant nodes doesn't always work well.



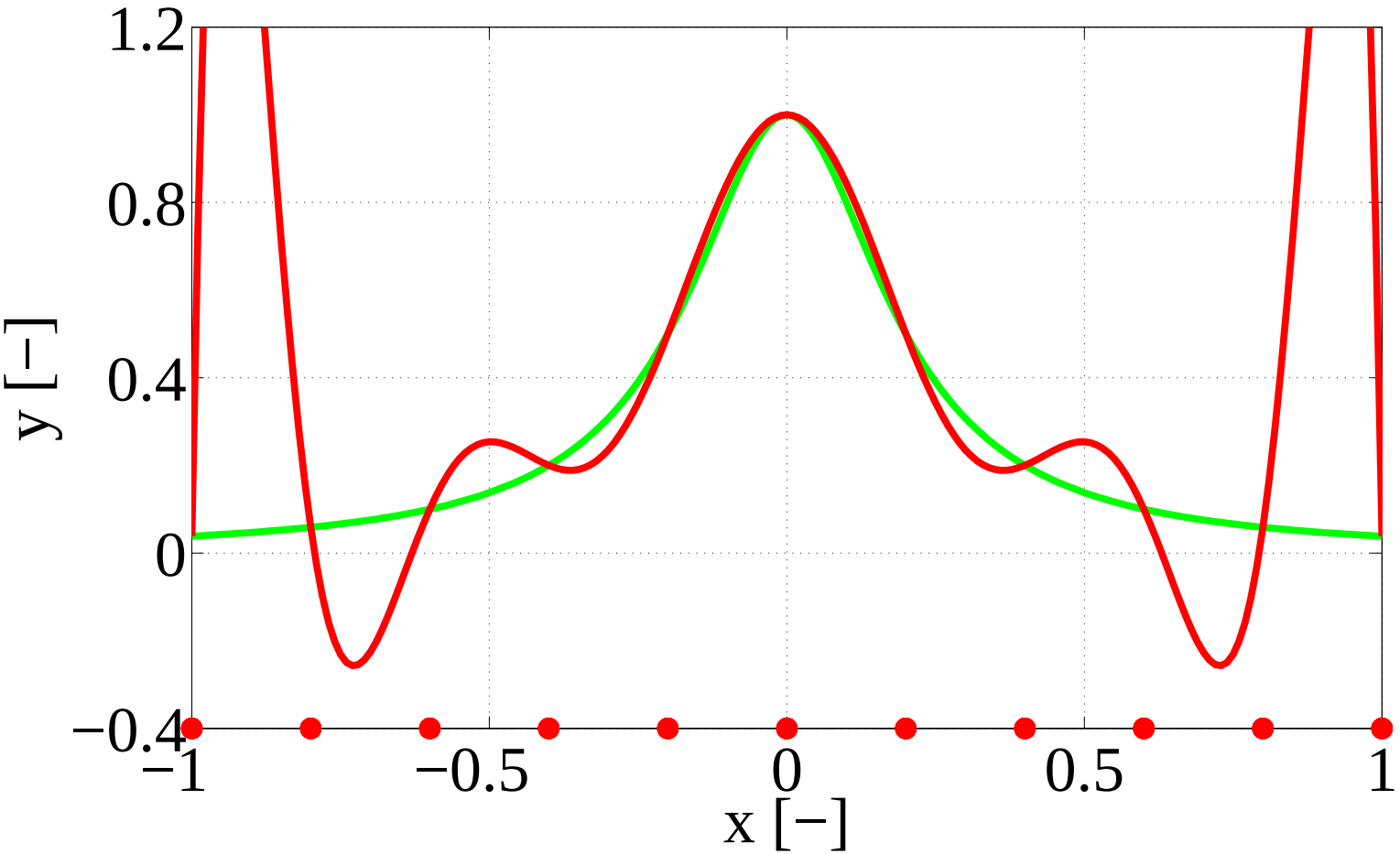
It's not so easy! Interpolation with equidistant nodes doesn't always work well.



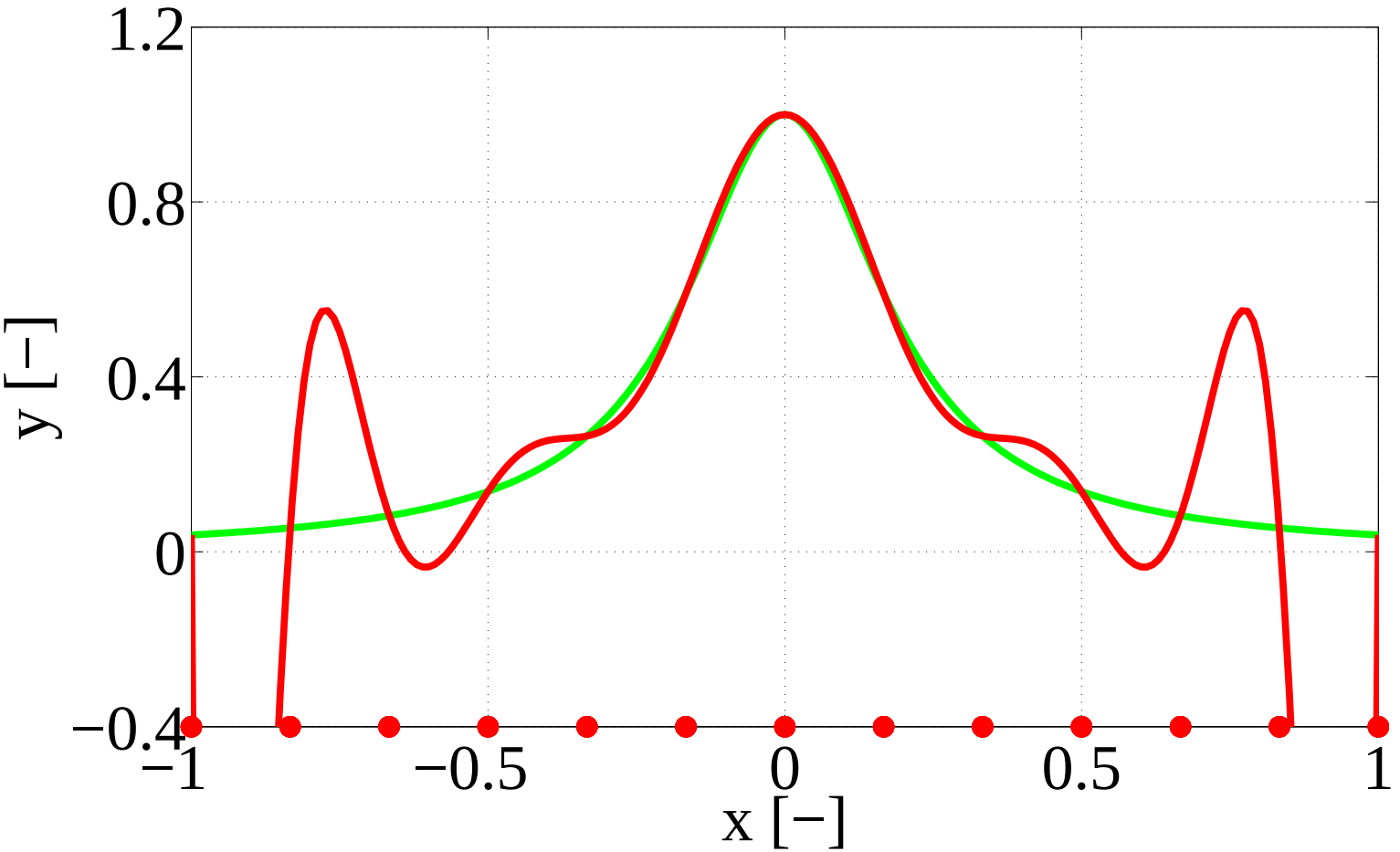
It's not so easy! Interpolation with equidistant nodes doesn't always work well.



It's not so easy! Interpolation with equidistant nodes doesn't always work well.

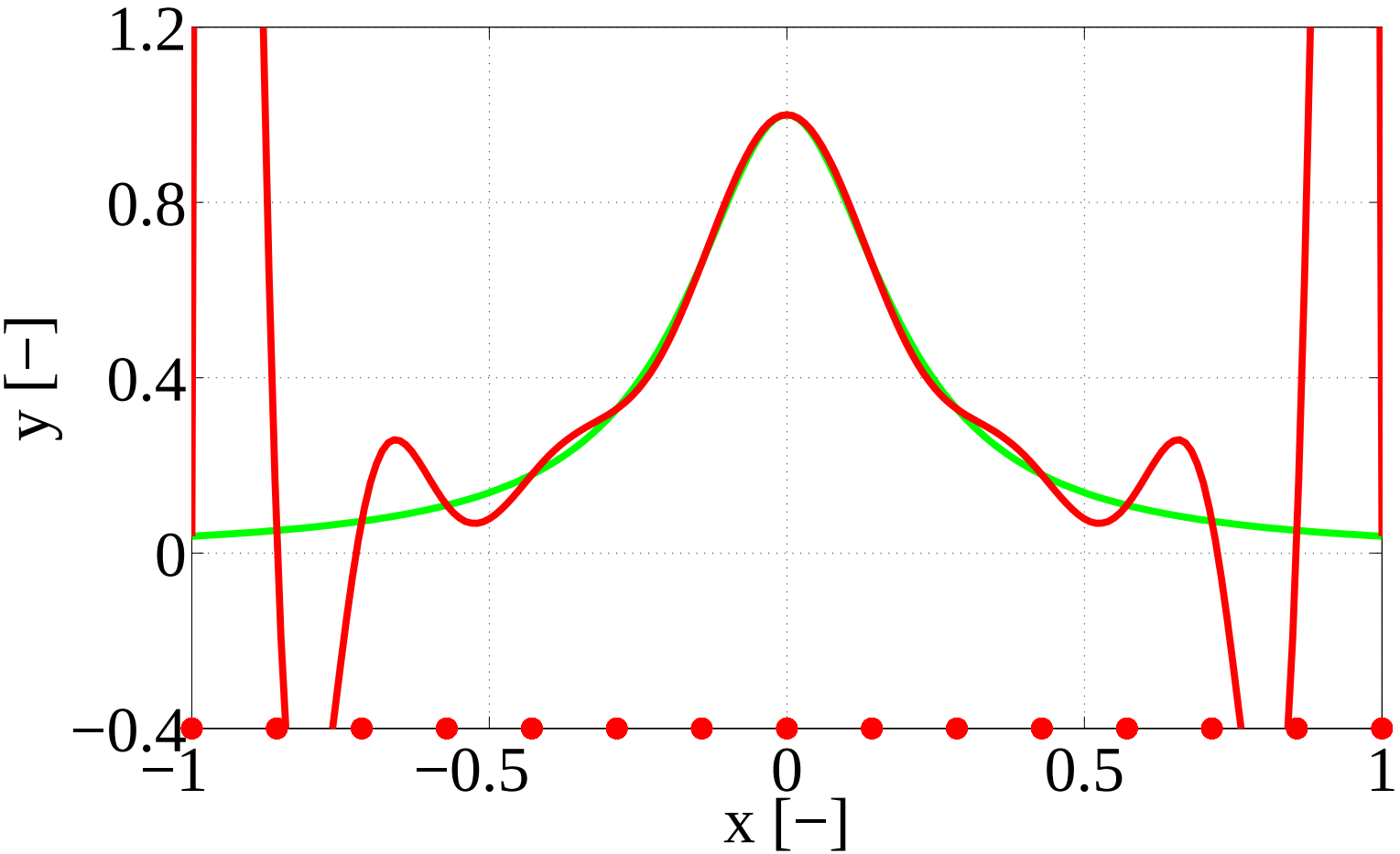


It's not so easy! Interpolation with equidistant nodes doesn't always work well.



p=2 p=4 p=6 p=8 p=10 **p=12** p=14

It's not so easy! Interpolation with equidistant nodes doesn't always work well.



It's not so easy! Interpolation with equidistant nodes doesn't always work well.

Outline

- Motivation.
- Outline.
- Setting.
- Nonintrusive stochastic projection method.
- Example.
- Conclusion.

Setting

- In this lecture, we assume that we deal with a problem of uncertainty quantification that involves a numerical simulation that is computationally costly to run.

- We will address the problem of constructing a surrogate model to serve as a computationally inexpensive substitute for the numerical simulation in the uncertainty quantification.

- A **surrogate model**, sometimes also called **metamodel**, is a model that mimics the relationship that the numerical simulation establishes between its input and output variables while being computationally less expensive.

- For the accessibility of the presentation, we will consider a problem with only a single input variable:

$$\underbrace{y}_{\text{output variable}} = \underbrace{g}_{\text{numerical simulation}} \left(\underbrace{x}_{\text{input variable}} \right) \quad \text{with} \quad g : \mathbb{R} \to \mathbb{R};$$

  we assume that the input variable has been assigned a probability density function ρ_X; in this context, we will describe in the following the well-known **nonintrusive stochastic projection method** for obtaining a surrogate model.

- The method can be extended to problems with many input variables, as we will see at the end.

Nonintrusive stochastic projection method

Least-squares polynomial approximation

- The nonintrusive stochastic projection method involves the use of a **polynomial surrogate model**.

- Conceptually, this polynomial surrogate model, which we denote by g^p where the superscript p is not an exponent but serves to distinguish the surrogate model from the numerical simulation as well as to indicate its polynomial degree, is obtained by seeking the degree-p polynomial that approximates the numerical simulation as accurately as possible in ρ_X**-weighted least-squares** sense:

$$\min_{c_0,c_1,\ldots,c_p} \frac{1}{2}\int_{\mathbb{R}} \left| g(x) - \sum_{\alpha=0}^{p} c_\alpha x^\alpha \right|^2 \rho_X(x)dx,$$

$$g^p(x) = \sum_{\alpha=0}^{p} c_\alpha x^\alpha = c_0 + c_1 x + \ldots + c_p x^p.$$

  The presence of the probability density function ρ_X can be expected to result in the approximation tending to be better for more highly probable values of the input variable.

- However, the relationship that the numerical simulation establishes between the input and output variables is typically not available in the form of a closed-form expression so that the integral in the objective function cannot be evaluated exactly, thus calling for a numerical integration.

Gauss quadrature

- A numerical integration, also called **quadrature**, is an approximation of an integral of a function over a domain of integration that is usually taken as $[-1, 1]$, usually in the form of a weighted sum of evaluations of this function at specified points, also called nodes, in the domain of integration.

- This notion of quadrature can be generalized by introducing a probability density function ρ_X, or conceivably a more general weight function, into the integrand and by allowing the domain of integration to be more general than $[-1, 1]$:

$$\int_{\mathbb{R}} f(x)\rho_X(x)dx \approx \sum_{\ell=1}^{\lambda} w_\ell f(x^{(\ell)}),$$

  in which $x^{(1)}, \ldots, x^{(\lambda)}$ denote the nodes and $w_1, \ldots, w_\lambda$ the corresponding weights. Please note that only the function itself is evaluated at the nodes and the presence of the probability density function is taken into account through the values of the nodes and the corresponding weights. Different choices of the probability density function lead to different quadrature rules.

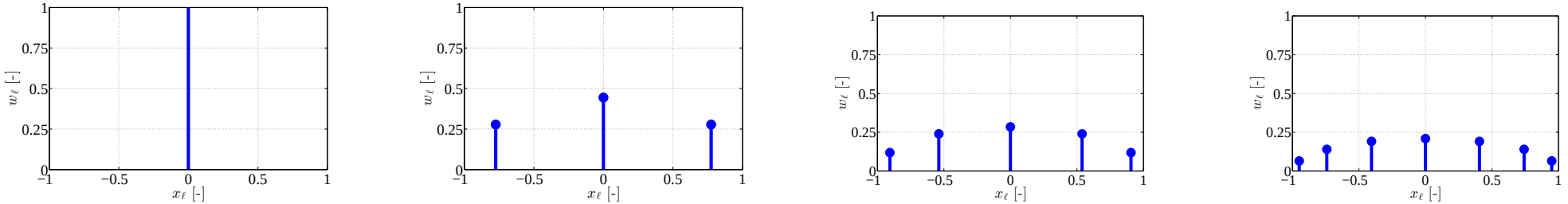
- A **λ-node Gauss quadrature for a PDF ρ_X** is a quadrature in which the nodes and weights are set up such that the quadrature is exact for all polynomials up to degree $2\lambda - 1$:

$$\int_{\mathbb{R}} x^\alpha \rho_X(x)dx = \sum_{\ell=1}^{\lambda} w_\ell (x^{(\ell)})^\alpha, \quad \alpha = 0, \ldots, 2\lambda - 1.$$

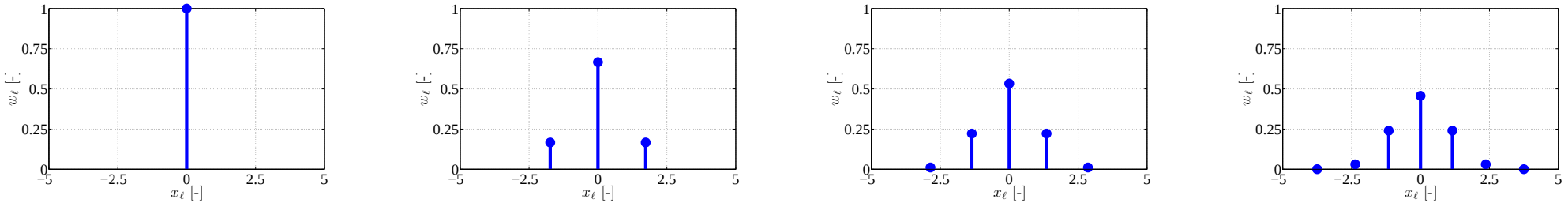
Gauss quadrature

- Gauss quadrature rules can be read from tables in the literature for many “labeled” probability density functions (uniform, Gaussian,. . .) or computed otherwise (by solving the nonlinear system of equations expressing the polynomial exactness to obtain the sought nodes and weights).

- Nodes and weights of Gauss quadratures with 1, 3, 5, and 7 nodes for integration with respect to PDF of uniform random variable with values in $[-1, 1]$, also called **Gauss–Legendre quadrature**:



- Nodes and weights of Gauss quadratures with 1, 3, 5, and 7 nodes for integration with respect to PDF of standard Gaussian random variable, also called **Gauss–Hermite quadrature**:



- Please note that the nodes are not equidistant.

Discrete least-squares polynomial approximation

- In the nonintrusive stochastic projection method, the **integral** in the objective function in the least-squares polynomial approximation problem is **approximated by means of Gaussian quadrature** to obtain the following discrete least-squares polynomial approximation problem:

$$\min_{c_0,c_1,\ldots,c_p} \frac{1}{2}\sum_{\ell=1}^{\lambda} w_\ell \left| g(x^{(\ell)}) - \sum_{\alpha=0}^{p} c_\alpha (x^{(\ell)})^\alpha \right|^2,$$

$$g^{p,\lambda}(x) = \sum_{\alpha=0}^{p} c_\alpha x^\alpha = c_0 + c_1 x + \ldots + c_p x^p,$$

in which the minimization problem can be rewritten equivalently as follows:

$$\min_{\boldsymbol{c}=(c_0,c_1,\ldots,c_p)} \frac{1}{2}(\boldsymbol{y} - [M]\boldsymbol{c})^{\mathrm{T}}[W](\boldsymbol{y} - [M]\boldsymbol{c}),$$

with

$$\boldsymbol{y} = \begin{bmatrix} g(x^{(1)}) \\ \vdots \\ g(x^{(\lambda)}) \end{bmatrix}, \quad [M] = \begin{bmatrix} 1 & (x^{(1)}) & (x^{(1)})^2 & \ldots & (x^{(1)})^p \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & (x^{(\lambda)}) & (x^{(\lambda)})^2 & \ldots & (x^{(\lambda)})^p \end{bmatrix}, \quad [W] = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_\lambda \end{bmatrix}$$

- Thus, in the nonintrusive stochastic projection method, the **numerical simulation must be run** for those values of the input variable that correspond to the **nodes of the Gaussian quadrature**.

Normal equations

- The equivalent reformulation involving matrices and vectors paves the way for a numerical solution by using **computational linear algebra** methods. The most widely used method is the **method of normal equations**, in which the expression of the stationarity of the objective function,

$$\frac{1}{2}(-[M]\delta\boldsymbol{c})^{\mathrm{T}}[W](\boldsymbol{y}-[M]\boldsymbol{c})+\frac{1}{2}(\boldsymbol{y}-[M]\boldsymbol{c})^{\mathrm{T}}[W](-[M]\delta\boldsymbol{c})=0, \quad \text{for all } \delta\boldsymbol{c} \text{ in } \mathbb{R}^{p+1},$$

leads to the so-called system of normal equations

$$[M]^{\mathrm{T}}[W][M]\boldsymbol{c}=[M]^{\mathrm{T}}[W]\boldsymbol{y},$$

which is then solved by using the Cholesky factorization of the system matrix as follows:

form the normal equations $[M]^{\mathrm{T}}[W][M]\boldsymbol{c}=[M]^{\mathrm{T}}[W]\boldsymbol{y}$;

compute the Cholesky factorization $[M]^{\mathrm{T}}[W][M]=[R]^{\mathrm{T}}[R]$;

first solve $[R]^{\mathrm{T}}\boldsymbol{d}=[M]^{\mathrm{T}}[W]\boldsymbol{y}$ and then solve $[R]\boldsymbol{c}=\boldsymbol{d}$.

Recall that the Cholesky factorization is a factorization of a positive definite matrix into the product of the transpose of an upper triangular matrix and this upper triangular matrix. And recall that a linear system involving a triangular matrix can be readily solved by using back-substitution.

- To ensure well-posedness (positive definiteness of $[M]^{\mathrm{T}}[W][M]$), the number of nodes in the Gauss quadrature must be greater than or equal to the number of coefficients to be determined, that is, $\lambda \geq p+1$. **Thus, when there is only a single input variable, the numerical simulation must be run at least $p+1$ times in order to fit a polynomial surrogate model of degree p to it.**

Polynomial chaos expansion

- It is interesting to analyze the aforementioned solution method more deeply because a deeper analysis allows to make a **link with an orthogonalization principle**.
- In fact, let us denote by $[A]$ the inverse of the upper triangular matrix $[R]$, that is,

$$[A] = [R]^{-1}.$$

The matrix $[A]$ is upper triangular. Since $[M]^{\mathrm{T}}[W][M] = [R]^{\mathrm{T}}[R]$, we have

$$[A]^{\mathrm{T}}[M]^{\mathrm{T}}[W][M][A] = [I],$$

in which $[I]$ is the identity matrix, an equation that can be rewritten equivalently as follows:

$$\sum_{\ell=1}^{\lambda} \Big(\sum_{\tilde{\alpha}=0}^{\alpha} (x^{(\ell)})^{\tilde{\alpha}} A_{\tilde{\alpha}\alpha} \Big) w_\ell \Big(\sum_{\tilde{\beta}=0}^{\beta} (x^{(\ell)})^{\tilde{\beta}} A_{\tilde{\beta}\beta} \Big) = \delta_{\alpha\beta}, \quad 0 \leq \alpha, \beta \leq p,$$

in which $\delta_{\alpha\beta}$ is the Kronecker delta equal to 1 if $\alpha = \beta$ and 0 otherwise. Assuming that $\lambda \geq p+1$, it follows from the polynomial exactness of the Gauss quadrature that

$$\int_{\mathbb{R}} \sum_{\tilde{\alpha}=0}^{\alpha} A_{\tilde{\alpha}\alpha} x^{\tilde{\alpha}} \sum_{\tilde{\beta}=0}^{\beta} A_{\tilde{\beta}\beta} x^{\tilde{\beta}} \rho_X(x) dx = \delta_{\alpha\beta}, \quad 0 \leq \alpha, \beta \leq p.$$

In conclusion, the **entries of $[A] = [R]^{-1}$**, that is, $\{A_{\alpha\beta},\ 0 \leq \alpha \leq \beta, 0 \leq \beta \leq p\}$, **define a collection of ρ_X-orthonormal polynomials of increasing degree** as follows:

$$\psi_\beta(x) = \sum_{\alpha=0}^{\beta} A_{\alpha\beta} x^\alpha \quad \text{with} \quad \int_{\mathbb{R}} \psi_\alpha(x) \psi_\beta(x) \rho_X(x) dx = \delta_{\alpha\beta}, \quad 0 \leq \alpha, \beta \leq p.$$

Polynomial chaos expansion

- Since $[R]\boldsymbol{c} = \boldsymbol{d}$, we have $c_\alpha = \sum_{\beta=0}^{p} A_{\alpha\beta} d_\beta$, $1 \leq \alpha \leq p$, and therefore

$$g^{p,\lambda}(x) = \sum_{\alpha=0}^{p} c_\alpha x^\alpha = \sum_{\alpha=0}^{p} \sum_{\beta=0}^{p} A_{\alpha\beta} d_\beta = \sum_{\beta=0}^{p} d_\beta \sum_{\alpha=0}^{p} A_{\alpha\beta} x^\alpha = \sum_{\beta=0}^{p} d_\beta \psi_\beta(x).$$

  This representation of the surrogate model, that is, its representation as an expansion in polynomials that are of increasing degree and orthonormal in the inner product weighted with respect to the probability density function of the input variable(s), is also called a **polynomial chaos expansion**. The solution of $[R]^{\mathrm{T}}\boldsymbol{d} = [M]^{\mathrm{T}}[W]\boldsymbol{y}$ provides precisely the coefficients $d_0, \ldots, d_p$.

- One of the advantages of the polynomial chaos expansion is that because the polynomials involved in it are of increasing degree and ρ_X-orthonormal and thus

$$\psi_0 = 1, \int_{\mathbb{R}} \psi_\beta(x)\rho_X(x)dx = 0, \ 1 \leq \beta \leq p, \int_{\mathbb{R}} \psi_\alpha(x)\psi_\beta(x)\rho_X(x)dx = \delta_{\alpha\beta}, \ 1 \leq \alpha, \beta \leq p,$$

  its coefficients provide approximations to the mean and the variance of the output variable:

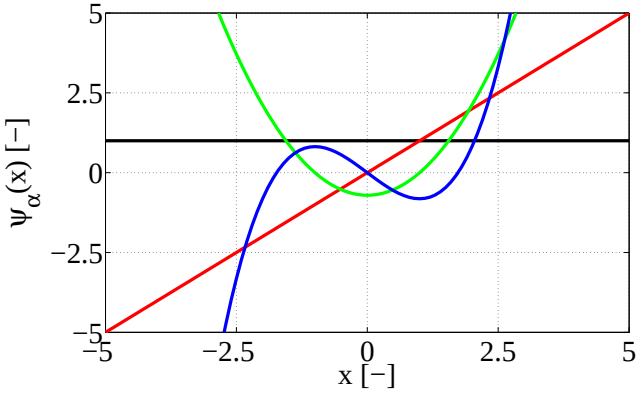
$$\overline{y} \approx \overline{y}^{p,\lambda} = \int_{\mathbb{R}} g^{p,\lambda}(x)\rho_X(x)dx = d_0,$$

$$\sigma_Y^2 \approx (\sigma_Y^{p,\lambda})^2 = \int_{\mathbb{R}} \left(g^{p,\lambda}(x) - \overline{y}^p\right)^2 \rho_X(x)dx = \sum_{\beta=1}^{p} d_\beta^2.$$

  In the multivariate case, the coefficients provide approximations to the variance-based sensitivity indices.

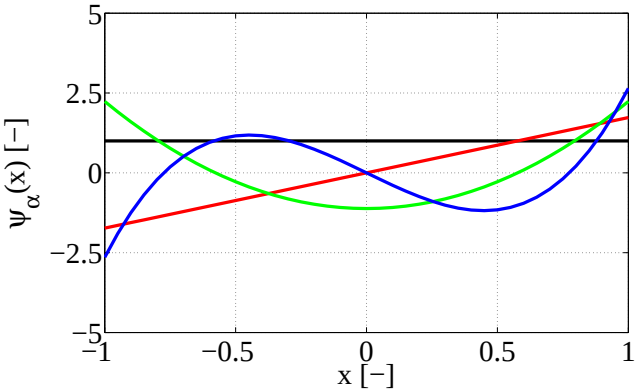
Polynomial chaos expansion

- Another advantage of the polynomial chaos expansion is that for certain "labeled" probability density functions (uniform, Gaussian,...), the sequence of polynomials that are of increasing degree and orthonormal in the inner product weighted with respect to it can be read from tables in the literature. Examples are cases wherein X is a standard Gaussian random variable (Hermite polynomials) or X is a uniform random variable with values in $[-1, 1]$ (Legendre polynomials):



Hermite polynomials

$$\psi_0 = 1$$
$$\psi_1 = x$$
$$\psi_2 = \frac{1}{\sqrt{2}}(x^2 - 1)$$



Legendre polynomials

$$\psi_0 = 1$$
$$\psi_1 = \sqrt{3}x$$
$$\psi_2 = \sqrt{5}\left(\tfrac{3}{2}x^2 - \tfrac{1}{2}\right)$$

- If a closed form expression is available for the polynomials $\psi_0, \ldots, \psi_p$, the coefficients $d_0, \ldots, d_p$ can be evaluated by rewriting the equation $[R]^{\mathrm{T}}\boldsymbol{d} = [M]^{\mathrm{T}}[W]\boldsymbol{y}$ equivalently as follows:

$$d_\beta = \sum_{\alpha=0}^{p} A_{\alpha\beta} \sum_{\ell=1}^{\lambda} (x^{(\ell)})^\alpha w_\ell g(x^{(\ell)}) = \sum_{\ell=1}^{\lambda} w_\ell g(x^{(\ell)}) \psi_\beta(x^{(\ell)}), \quad 0 \le \beta \le p.$$

Remarks

- The numerical solution of discrete least-squares polynomial approximation problems can be sensitive to numerical errors (notorious issue of ill-conditioning of the Vandermonde matrix, the normal equations,...). The accuracy of the numerical solution can be improved through a **normalization** that involves constructing the surrogate model in terms of a normalized input variable obtained by subtracting from the input variable its mean and dividing by its standard deviation:

$$\min_{c_0,c_1,\ldots,c_p} \frac{1}{2}\sum_{\ell=1}^{\lambda} w_\ell \left| g(x^{(\ell)}) - \sum_{\alpha=0}^{p} c_\alpha \left(\frac{x^{(\ell)} - \overline{x}}{\sigma_X} \right)^\alpha \right|^2,$$

$$g^{p,\lambda}(x) = \sum_{\alpha=0}^{p} c_\alpha x^\alpha = c_0 + c_1 \left(\frac{x - \overline{x}}{\sigma_X} \right) + \ldots + c_p \left(\frac{x - \overline{x}}{\sigma_X} \right)^p.$$

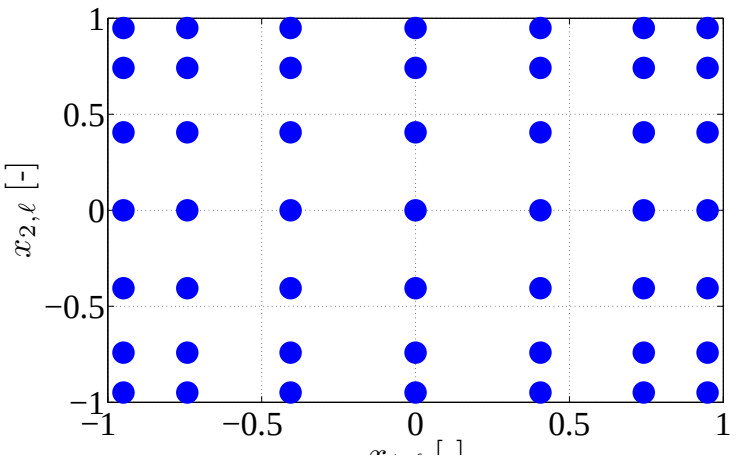
- An alternative method for obtaining a polynomial surrogate model is the so-called **stochastic collocation method**, which uses Lagrange polynomial interpolation:

$$g^\lambda(x) = \sum_{\ell=1}^{\lambda} g(x^{(\ell)}) l_\ell(x), \quad l_\ell(x) = \prod_{\substack{1 \le k \le \lambda \\ k \ne \ell}} \frac{x - x^{(k)}}{x^{(\ell)} - x^{(k)}}, \quad 1 \le \ell \le \lambda,$$

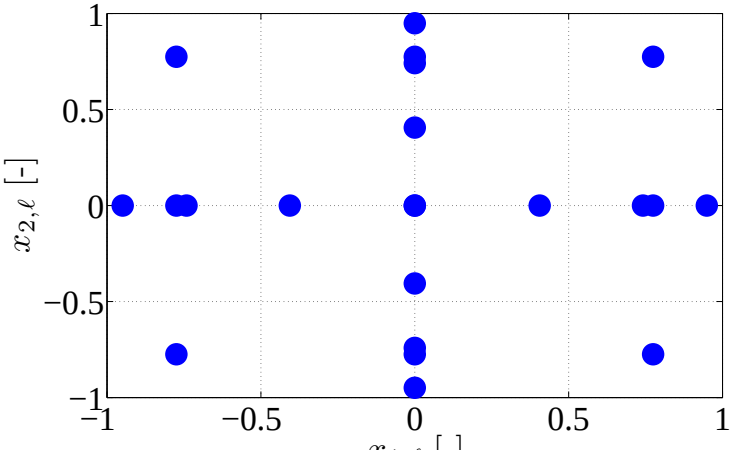
where $x^{(1)}, \ldots, x^{(\lambda)}$ still denote the nodes of the λ-node Gauss quadrature. When there is only a single input variable and a Gauss quadrature with $\lambda = (p+1)$ nodes is used in order to fit a polynomial surrogate model of degree p, the nonintrusive stochastic projection method and the stochastic collocation method coincide (up to errors stemming from the numerical implementation).

Extension to the multivariate case

- The nonintrusive stochastic projection method and the stochastic collocation method can be extended to problems involving multiple input variables through the use of multivariate polynomials and the use of tensorized Gauss quadrature:



Full tensorization.



Sparse-grid tensorization.

Thus, when there are d input variables and a fully tensorized Gauss quadrature is used, the numerical simulation must be run at least $(p+1)^d$ times in order to fit a polynomial surrogate model of degree p to it. We can see that in the multivariate case, the computational cost of the construction of the surrogate model grows exponentially with the number of input variables. This issue is known in multivariate approximation theory as the so-called **curse of dimensionality**.

- Within the present state of the art, the Monte Carlo method is used for problems with high number of input variables (high dimension) and stochastic expansion methods involving surrogate models for problems with low to moderate number of input variables (low to moderate dimension).

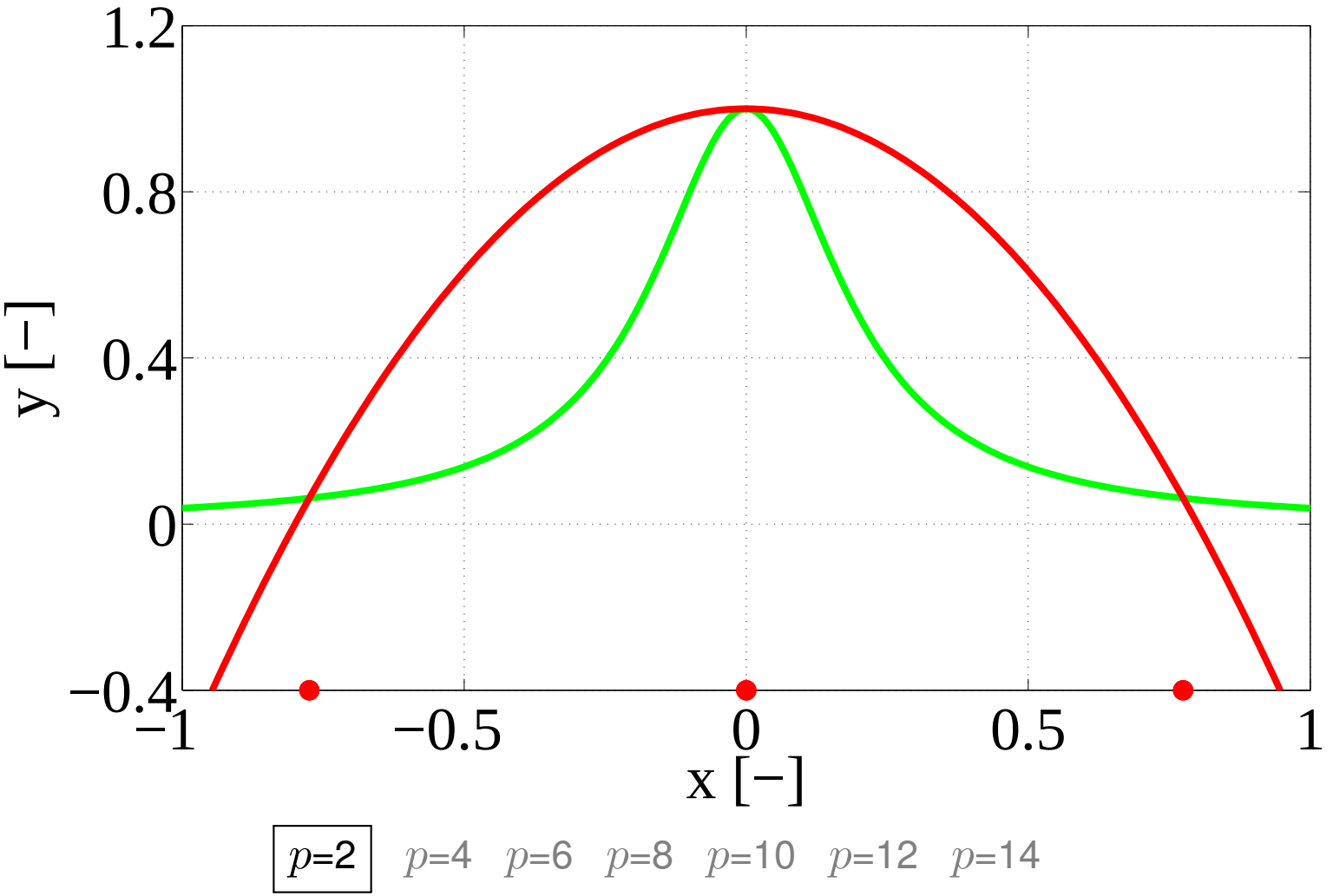
Example

Example

- Simple implementation for $g(x) = 1/(1+25x^2)$ and ρ_X PDF of uniform r.v. with values in $[-1, 1]$:

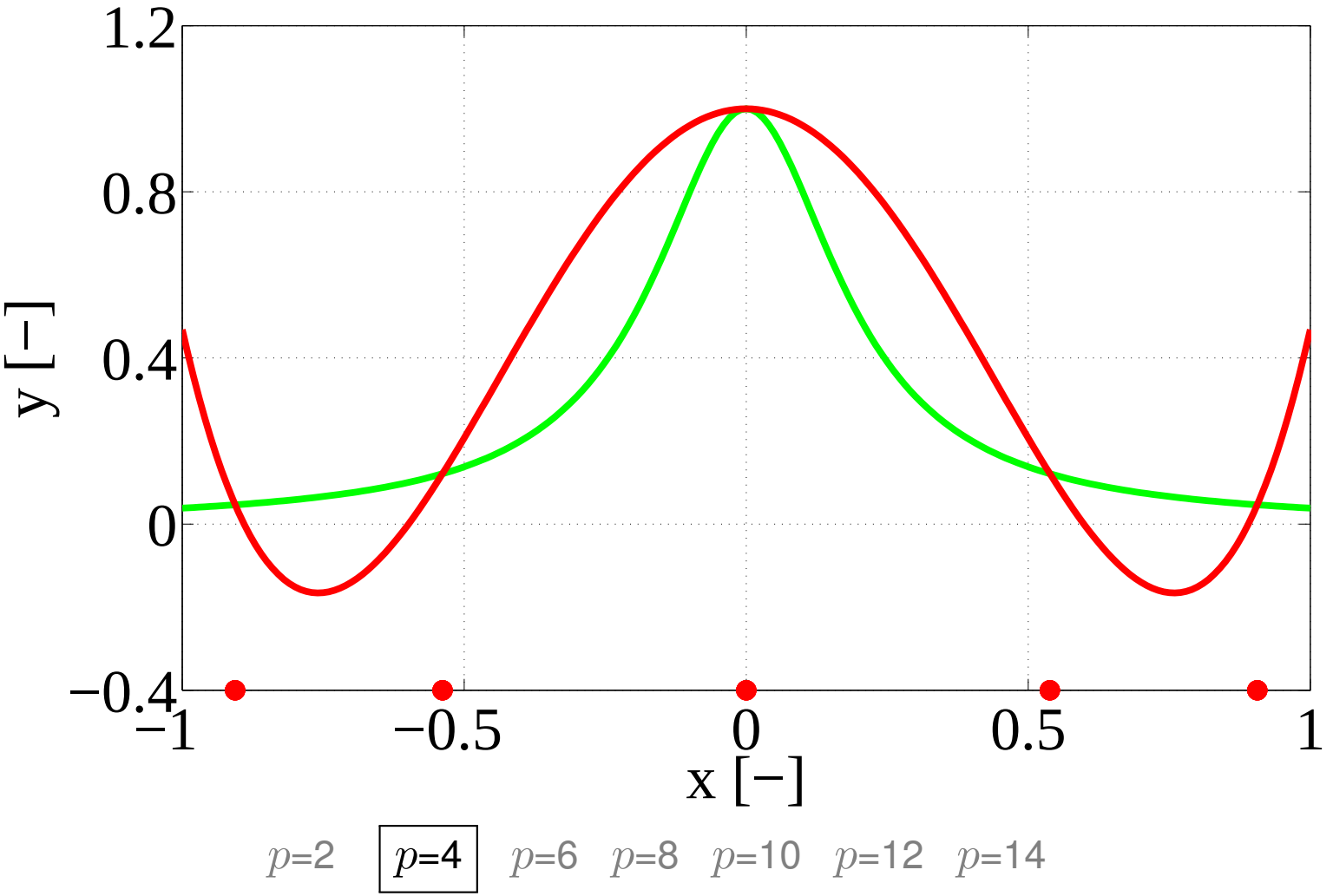
```
p=2; % lambda=p+1

% https://en.wikipedia.org/wiki/Gaussian_quadrature
xell=[-sqrt(3/5);0;sqrt(3/5)];
well=[2.5/9;4/9;2.5/9];

M=repmat(xell,[1 p+1]).^repmat([0:p], [p+1 1]);
G=M'*diag(well)*M;
R=chol(G);
d=R'\(M'*diag(well)*(1./(1+25*xell.^2)));
c=R\d;

figure;hold on;
plot([-1:0.01:1],1./(1+25*[-1:0.01:1].^2),'g-');
plot([-1:0.01:1],polyval(c(end:-1:1),[-1:0.01:1]),'r-');
```

Example



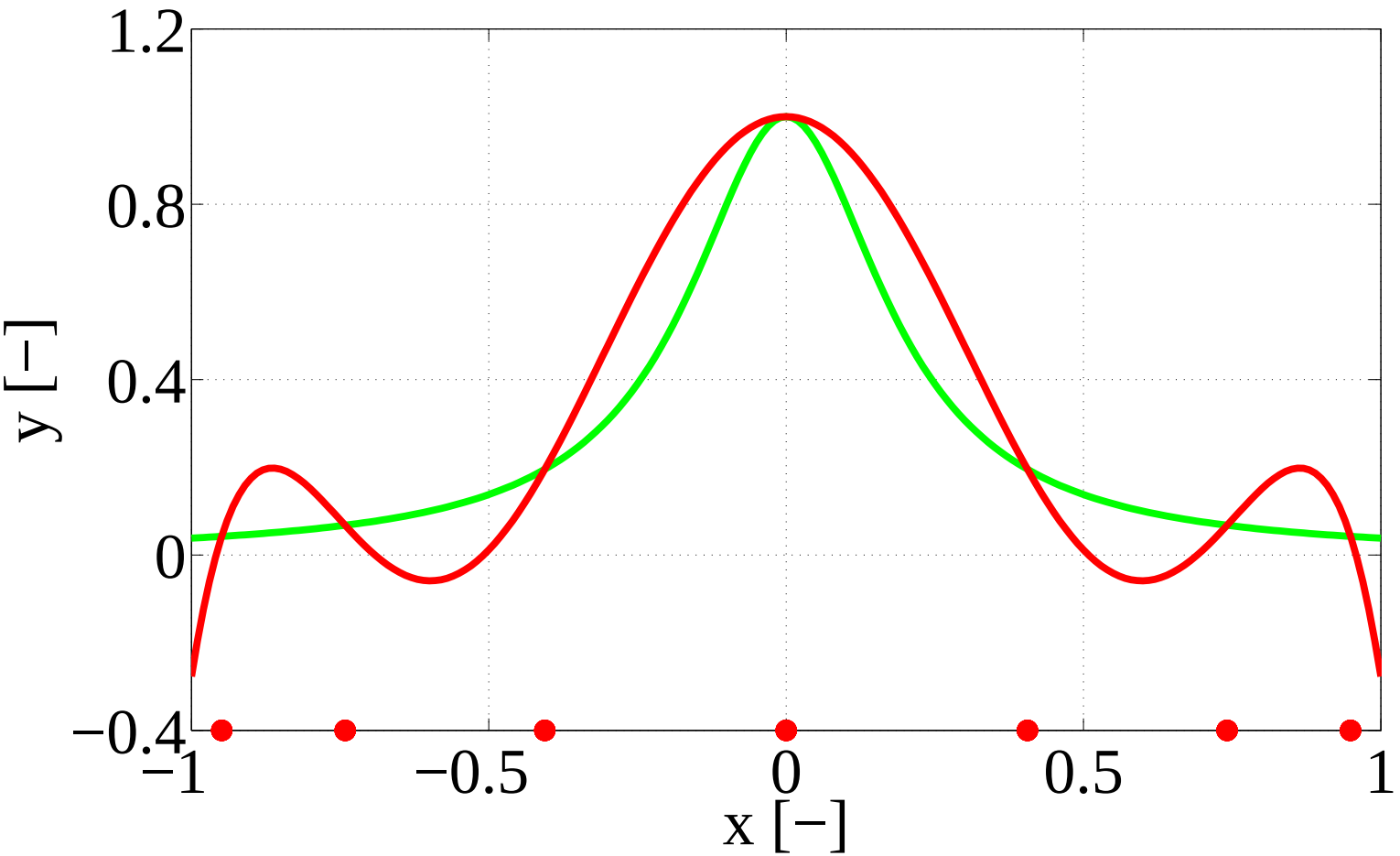
Nonintrusive stochastic projection method.

Example



Nonintrusive stochastic projection method.

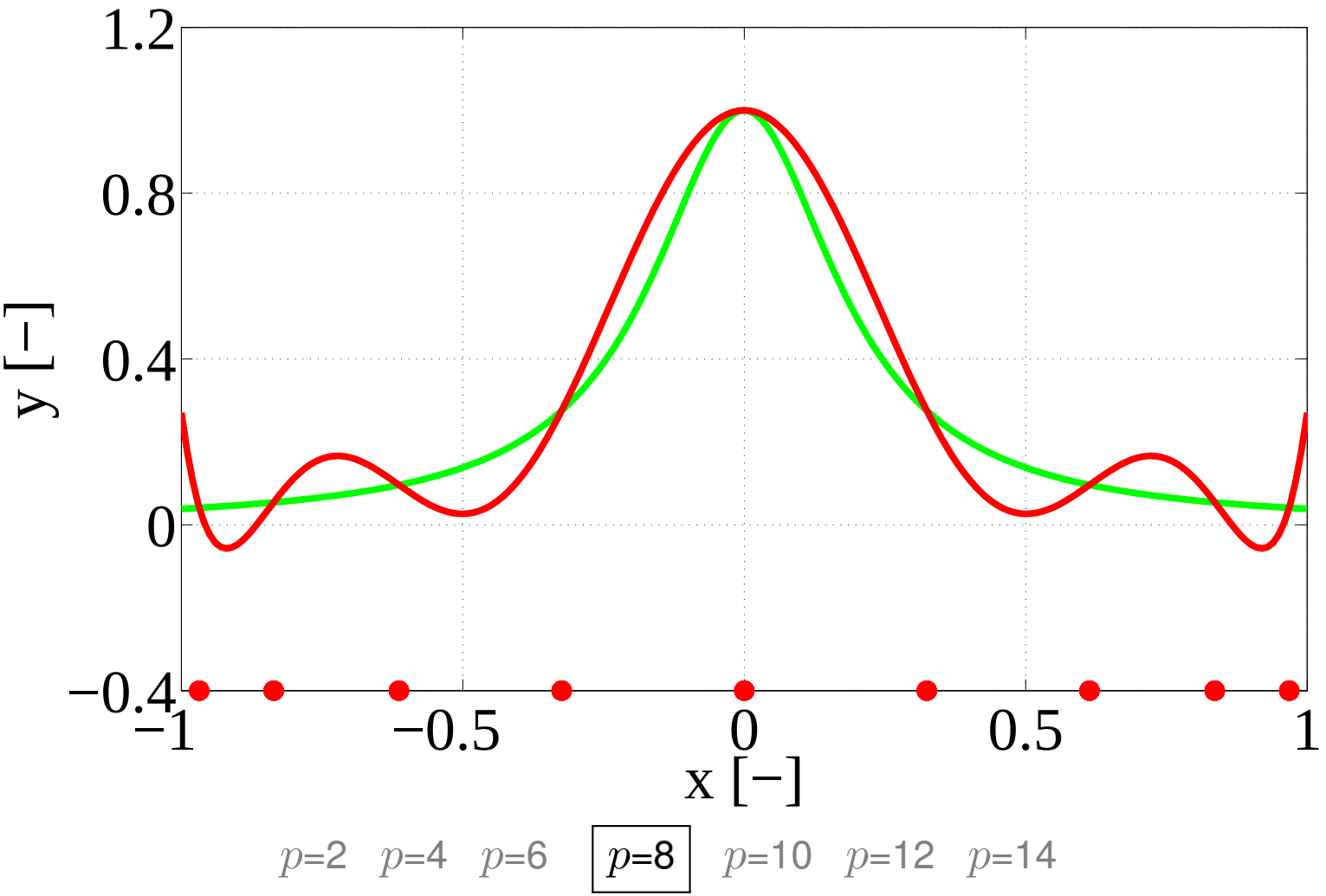
Example



p=2 p=4 **p=6** p=8 p=10 p=12 p=14

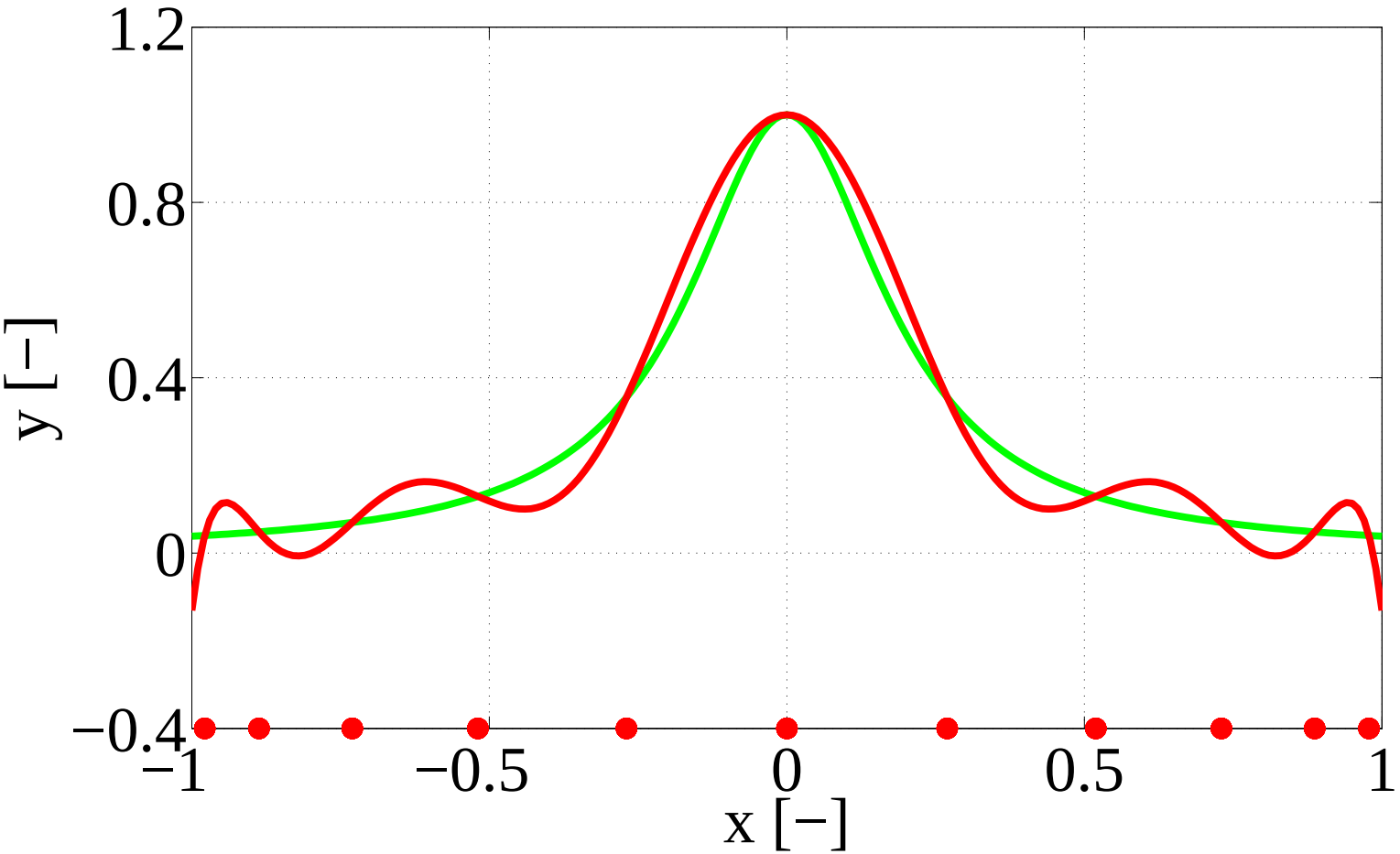
Nonintrusive stochastic projection method.

Example



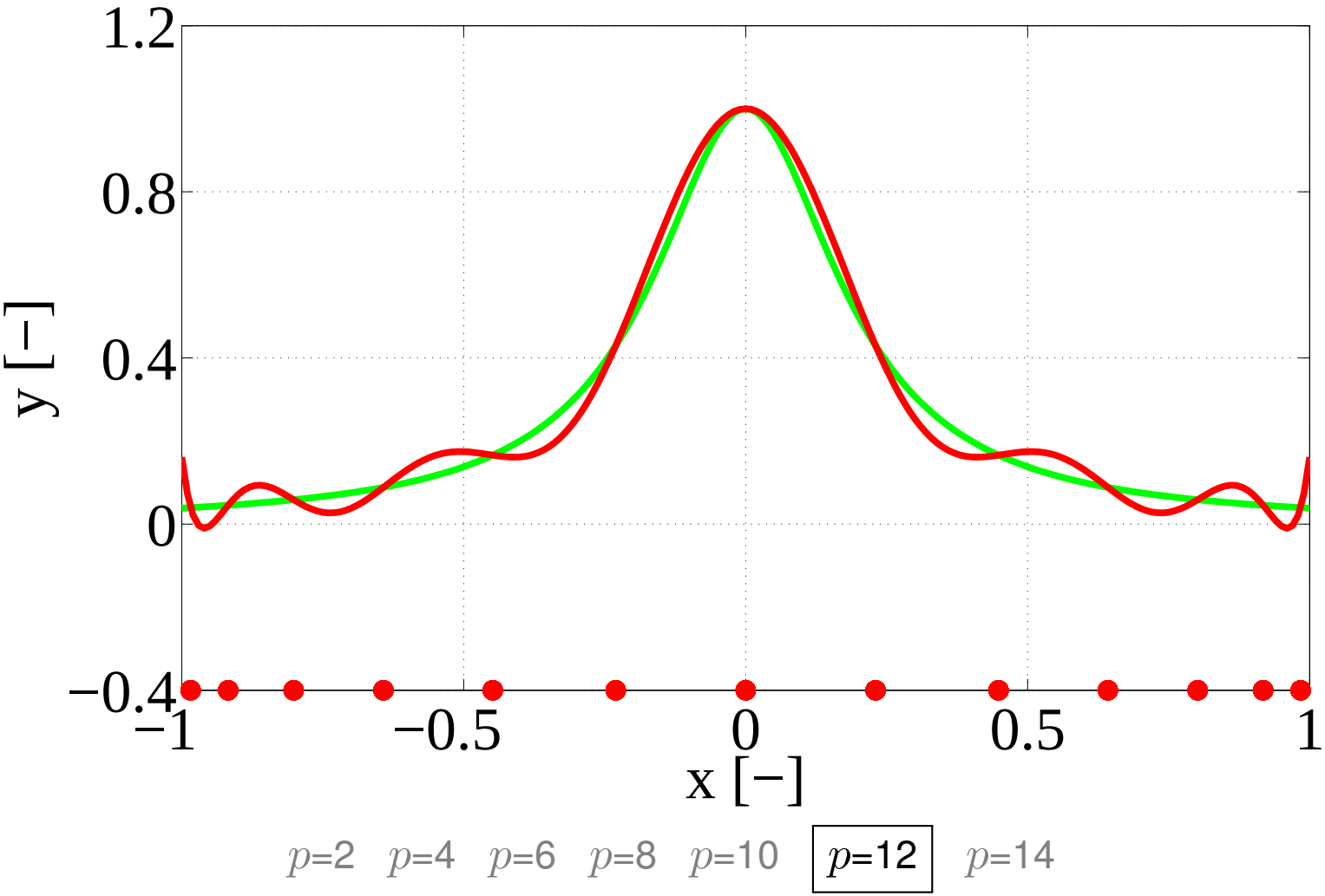
Nonintrusive stochastic projection method.

Example



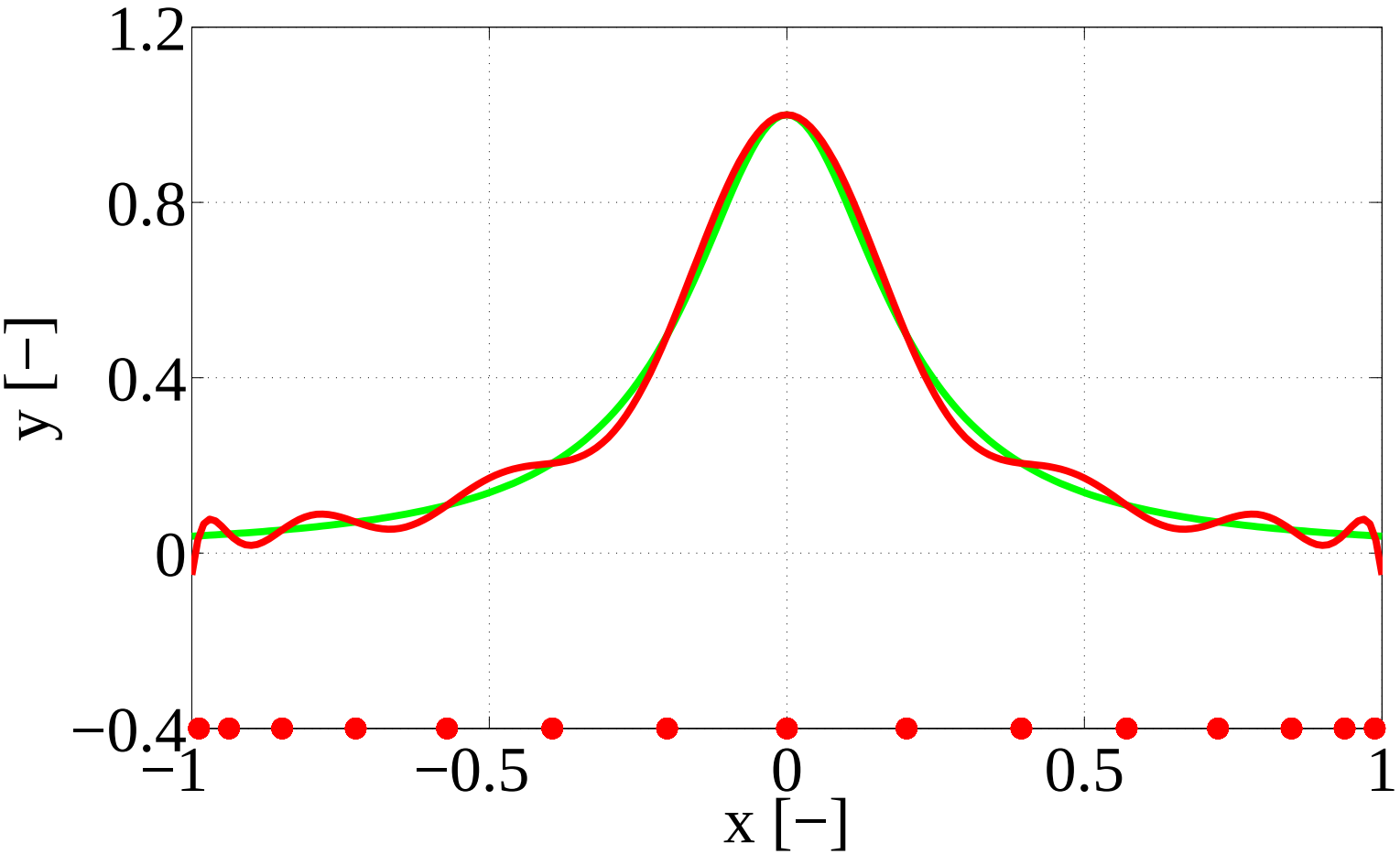
Nonintrusive stochastic projection method.

Example



Nonintrusive stochastic projection method.

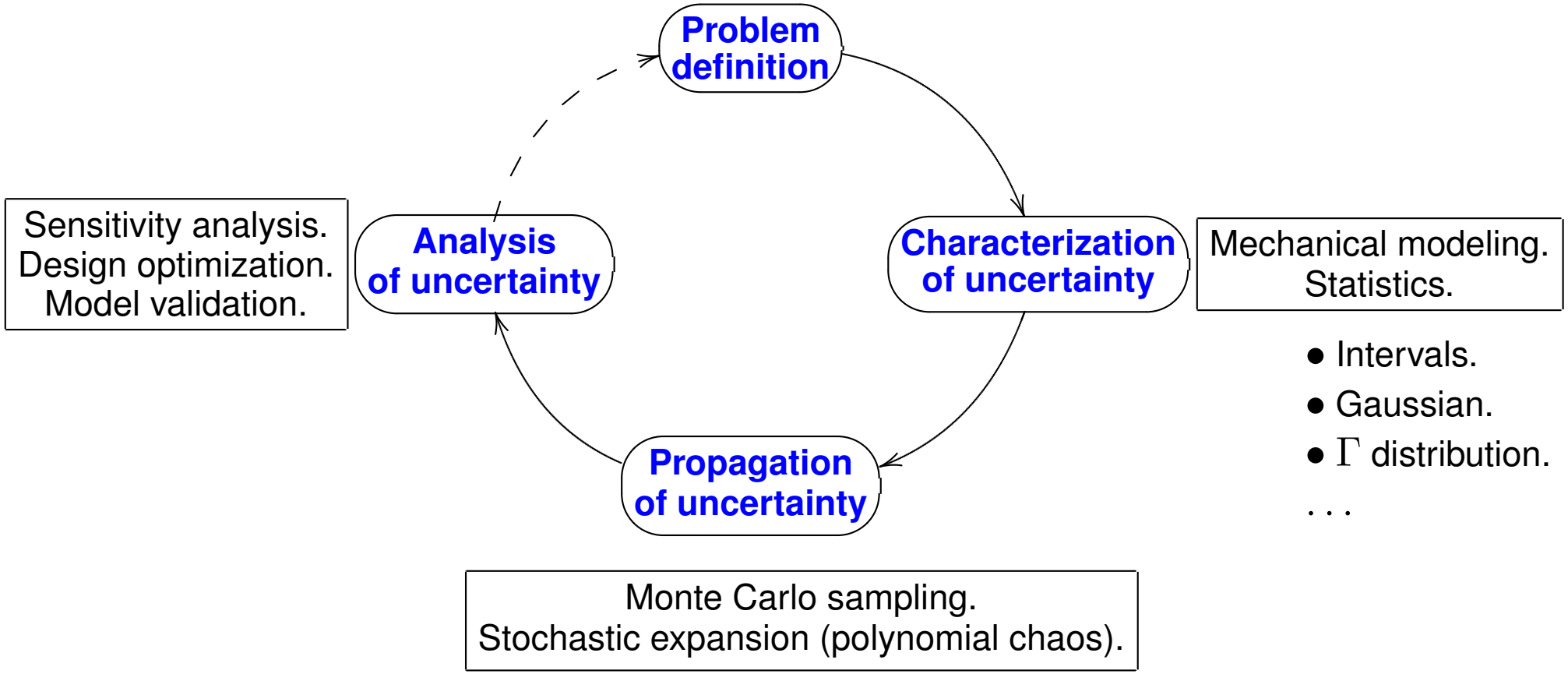
Example



Nonintrusive stochastic projection method.

Conclusion

Here we are now...



The computational cost of stochastic methods can be lowered
via the use of a surrogate model as a substitute for a numerical simulation or real tests.

References

- C. Dunkl and Y. Xu. Orthogonal Polynomials of Several Variables, 2001.
- R. Ghanem and P. Spanos. Stochastic Finite Elements: A Spectral Approach, 2003.
- G. Golub and C. Van Loan. Matrix computations, 1996.
- G. Golub and G. Meurant. Matrices, Moments and Quadrature with Applications, 2009.
- M. Holtz. Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance, 2010.
- O. Le Maître and O. Knio. Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics, 2010.
- J. Nocedal and S. Wright. Numerical Optimization, 2006.
- D. Xiu. Numerical Methods for Stochastic Computations: A Spectral Method Approach, 2010.
- M. Arnst and J.-P. Ponthot. An overview of nonintrusive characterization, propagation, and sensitivity analysis of uncertainties in computational mechanics. International Journal for Uncertainty Quantification, 4:387–421, 2014.