UNIVERSITY OF Southampton
School of Engineering Sciences

Conjugate gradient methods for parametrized linear random algebraic equations

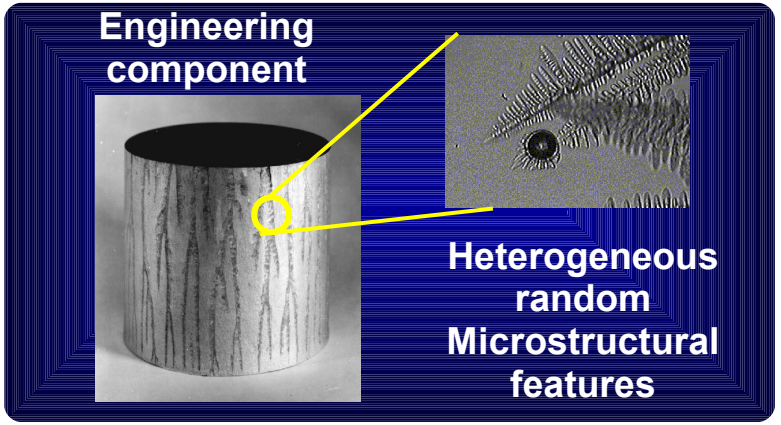
Dr Pär Håkansson & Dr Prasanth B. Nair
Computational Engineering and Design Group
27h August 2009

Talk Outline

- Introduction
- Steady state stochastic diffusion problem
- Formulate algorithm within CG framework
- Conventional Stochastic Galerkin approach
- Results & new insights
- Future developments

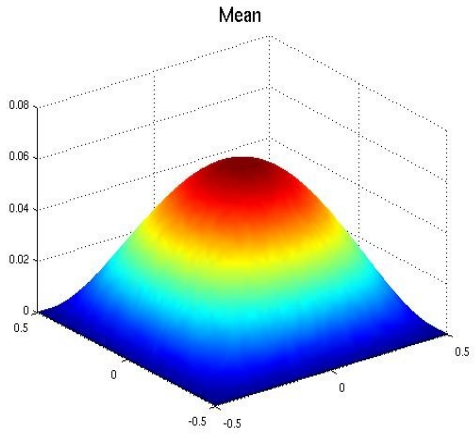
Uncertainty in engineering and natural systems

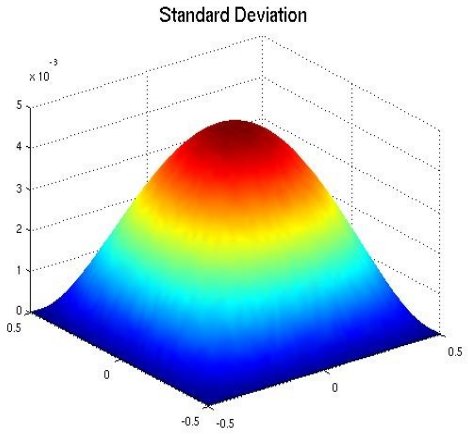
- Random field accounting for intrinsic randomness or ignorance in building model
- Solve time independent SPDE
- Output: mean and higher moments to characterize



URL: http://mpdc.mae.cornell.edu/

Example diffusion in heterogeneous media:





Random field as truncated Fourier-like expansion

$$\kappa(x,\omega) = 1 + \sigma \sum_{i=1}^{M} \xi_i \kappa_i(x)$$

First order in M independent random variables ξ_i

Karhunen-Loéve Expansion

Output as Polynomial Chaos Expansion (Wiener,Karniadakis,Ghanem)

$$w(\xi) = \sum_{i=0}^{P_w} w_i \varphi_i(\xi) \quad \text{with } w_i = \frac{\langle \varphi_i(\xi) w(\xi) \rangle}{\langle \varphi_i(\xi)^2 \rangle}$$

$$P_w + 1 = \frac{(M + p_w)!}{M! p_w!}$$

$\{\varphi_i(\xi)\}$ orthogonal PC basis function of order p_w or lower.

Legendre, Hermite, etc

Test problem stochastic diffusion equation [1]:

$$\begin{aligned} \nabla.[\kappa(x,\omega)\nabla u(x,\omega)] &= f(x) \text{ in } \mathcal{D}\times\Omega \\ u(x,\omega) &= g(x) \text{ on } \partial\mathcal{D}\times\Omega \end{aligned}$$

Homogeneous boundary condition (g). Continuous random field in KL-expansion(unit correlation length and size of domain):

$$\kappa(x,\omega) = 1 + \sigma \sum_{i=1}^{M} \xi_i \kappa_i(x)$$

Truncated at M terms,
Standard deviation σ
ξ_i Gaussian random variables

FEM on domain finally provides: $\underset{n\,x\,n}{A(\xi)}\ \underset{n}{x(\xi)} = b(\xi)$

[1] C. Powell and C. Elman, Block-diagonal preconditioning for spectral stochastic finite elements systems, IMA J. Num. Anal., **29**, pp 350-375 (2009)

UNIVERSITY OF Southampton
School of Engineering Sciences

Can we derive a new algorithm when A is *a.s.* SPD?

$$A(\xi)x(\xi) = b(\xi)$$

Almost surely SPD: $A(\xi)^T = A(\xi) \quad \text{and} \quad x^T A(\xi)x > 0 \; \forall \; x \neq 0$

True solution: $x^*(\xi) = A(\xi)^{-1}b(\xi)$

Define A-norm: $||x(\xi)||_A^2 = \int_\Gamma x(\xi)^T A(\xi)x(\xi)\mathcal{P}(\xi)\mathrm{d}\xi$

Parametrized Krylov subspace:

$$\mathcal{K}_k(A(\xi), r_0) = \text{span}\left(r_0, A(\xi)r_0, \ldots, A(\xi)^{k-1}r_0\right) \quad \text{for} \quad k \geq 1$$

where $r_0 = b(\xi) - A(\xi)x_0$.

Prototype CG algorithm

Minimize cost function: $\phi(x) = \frac{1}{2}\langle x(\xi)^T A(\xi) x(\xi)\rangle - \langle x(\xi)^T b(\xi)\rangle$

A-norm: $||x(\xi) - x^*(\xi)||_A^2$

orthogonal residuals: $\langle r_k(\xi)^T r_l(\xi)\rangle = 0$ iterations $l < k$

theorems on the update of

$x(\xi)$ $r(\xi)$ $p(\xi)$

ALGORITHM 4.1: sCG$(x(\xi), b(\xi), A(\xi), \epsilon, k_{max})$

1. $r(\xi) = b(\xi) - A(\xi)x(\xi)$, $\rho^1 = \langle r(\xi)^T r(\xi)\rangle$, $k = 1$.
2. Do While $\sqrt{\rho^k} > \epsilon\sqrt{\langle b(\xi)^T b(\xi)\rangle}$ and $k < k_{max}$
   - (a) if $k = 1$ then $p(\xi) = r(\xi)$ else $\beta = \rho^k/\rho^{k-1}$ and $p(\xi) = r(\xi) + \beta p(\xi)$
   - (b) $w(\xi) = A(\xi)p(\xi)$
   - (c) $\alpha = \rho^k/\langle p(\xi)^T w(\xi)\rangle$
   - (d) $x(\xi) = x(\xi) + \alpha p(\xi)$
   - (e) $r(\xi) = r(\xi) - \alpha w(\xi)$
   - (f) $\rho^{k+1} = \langle r(\xi)^T r(\xi)\rangle$
   - (g) $k = k + 1$

To implement prototype algorithm we need:

a) Inner product between two PC vectors

$$\langle x(\xi)^T y(\xi) \rangle = \sum_{i=0}^{P} x_i^T y_i \langle \varphi_i^2 \rangle,$$

b) PC – Matrix vector operation

$$y_i = \frac{1}{\langle \varphi_i^2 \rangle} \sum_{j=0}^{P_A} \sum_{k=0}^{P_x} \langle \varphi_i \varphi_j \varphi_k \rangle A_j x_k, \quad \text{for } i = 0, 1, 2, \ldots, P_y$$

Practical implementation, PC variables

ALGORITHM 4.3: PC-sPCG$(x(\xi), b(\xi), A(\xi), M, \epsilon, k_{max}, P_x)$

1. Compute $r_i = b_i - (1/\langle\varphi_i^2\rangle)\sum_{j=0}^{P_A}\sum_{l=0}^{P_x}\langle\varphi_i\varphi_j\varphi_l\rangle A_j x_l$, for $i = 0, 1, 2, \ldots, P_x$
   $\rho^1 = \sum_{i=0}^{P_x} r_i^T r_i \langle\varphi_i^2\rangle$, $k = 1$.
2. Do While $\sqrt{\rho^k} > \epsilon\sqrt{\sum_{i=0}^{P_b} b_i^T b_i \langle\varphi_i^2\rangle}$ and $k < k_{max}$

   (a) $z_i = M r_i$, for $i = 0, 1, 2, \ldots, P_x$
   (b) $\tau^k = \sum_{i=0}^{P_x} z_i^T r_i \langle\varphi_i^2\rangle$
   (c) if $k = 1$ then $\beta = 0$ and $p_i = z_i$ for $i = 0, 1, 2, \ldots, P_x$
   else $\beta = \tau^k/\tau^{k-1}$ and $p_i = z_i + \beta p_i$, for $i = 0, 1, 2, \ldots, P_x$
   (d) $w_i = (1/\langle\varphi_i^2\rangle)\sum_{j=0}^{P_A}\sum_{l=0}^{P_x}\langle\varphi_i\varphi_j\varphi_l\rangle A_j p_l$, for $i = 0, 1, 2, \ldots, P_x$
   (e) $\alpha = \tau^k/(\sum_{i=0}^{P_x} p_i^T w_i \langle\varphi_i^2\rangle)$
   (f) $x_i = x_i + \alpha p_i$, for $i = 0, 1, 2, \ldots, P_x$
   (g) $r_i = r_i - \alpha w_i$, for $i = 0, 1, 2, \ldots, P_x$
   (h) $\rho^{k+1} = \sum_{i=0}^{P_x} r_i^T r_i \langle\varphi_i^2\rangle$
   (i) $\tau_{k-1} = \tau_k$
   (j) $k = k + 1$

Preconditioners: Cholesky (complete/incomplete) of $\langle A(\xi)\rangle$

UNIVERSITY OF Southampton
School of Engineering Sciences

Computational complexity

1 PC-Matrix vector $w=Ap$ (dominate)

2 Preconditioner and PC-scalar products

Memory requirement

PC-variables: x,r,w and p ($4 \times P_x \times n$)

Refer to Stochastic Galerkin approach (Ghanem and Spanos)

1) Introduce PC-variables:

Galerkin projection to determine x_i

$$\left\langle \varphi_j \left(A(\xi) \sum_{i=0}^{P_x} x_i \varphi_i(\xi) - b(\xi) \right) \right\rangle = 0, \quad \text{for } j = 0, 1, \ldots, P_x$$

Large deterministic linear equation (not completely assembled)

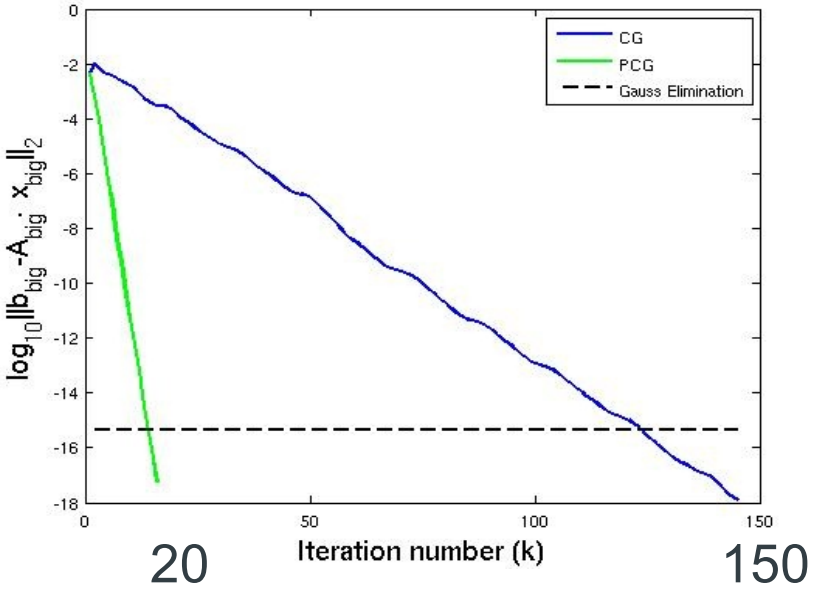
$$\underbrace{\sum_{i=0}^{P_A} \begin{bmatrix} \langle \varphi_i \varphi_0^2 \rangle A_i & \langle \varphi_i \varphi_0 \varphi_1 \rangle A_i & . & \langle \varphi_i \varphi_0 \varphi_{P_x} \rangle A_i \\ \langle \varphi_i \varphi_1 \varphi_0 \rangle A_i & \langle \varphi_i \varphi_1^2 \rangle A_i & . & \langle \varphi_i \varphi_1 \varphi_{P_x} \rangle A_i \\ . & . & . & . \\ . & . & . & . \\ \langle \varphi_i \varphi_{P_x} \varphi_0 \rangle A_i & \langle \varphi_i \varphi_{P_x} \varphi_1 \rangle A_i & . & \langle \varphi_i \varphi_{P_x}^2 \rangle A_i \end{bmatrix}}_{A_{big}} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ x_{P_x} \end{bmatrix}}_{x_{big}} = \underbrace{\begin{bmatrix} \langle \varphi_0^2 \rangle b_0 \\ \langle \varphi_1^2 \rangle b_1 \\ . \\ . \\ \langle \varphi_P^2 \rangle b_{P_x} \end{bmatrix}}_{b_{big}}$$

2) Solve with CG, block diagonal preconditioned

Algorithm 4.3 mathematically equivalent to Stochastic Galerkin for fixed P_x !

UNIVERSITY OF Southampton
School of Engineering Sciences

Complication in Stochastic Galerkin and Algorithm 4.3



20 150

Test case:

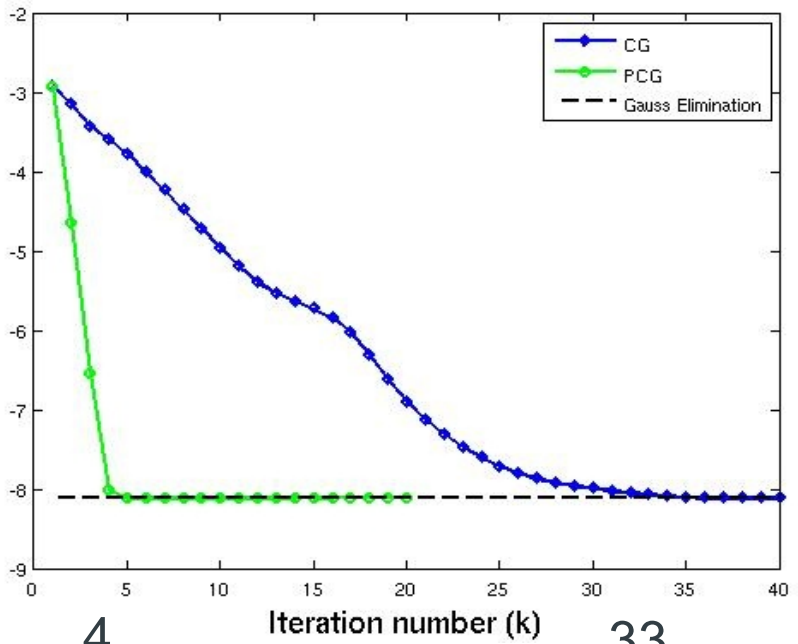
p_x=3 (PC-order of solution x)

M=4 (Dimension of random field)

σ=0.1

L_2 error norm

(x* computed with p_x=6 quadrature)



Wasted iterations!

4 33

True residual

UNIVERSITY OF Southampton
School of Engineering Sciences

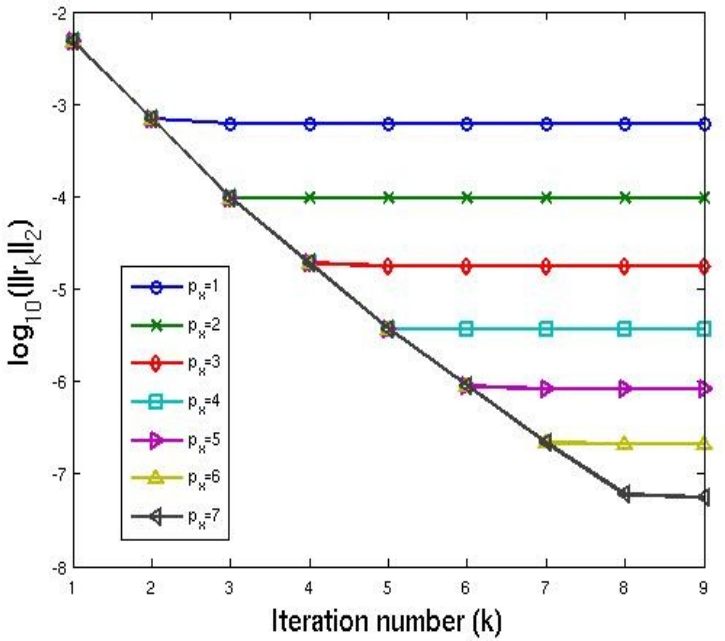
$$r(\xi) = b(\xi) - A(\xi)x(\xi)$$

$$r_i = b_i - (1/\langle\varphi_i^2\rangle)\sum_{j=0}^{P_A}\sum_{l=0}^{P_x}\langle\varphi_i\varphi_j\varphi_l\rangle A_j x_l$$

in PC-basis $p_A + p_x$

$$||r||_2^2 = \underbrace{\sum_{i=0}^{P_x} r_i^T r_i\langle\varphi_i^2\rangle}_{o(p_x)} + \underbrace{\sum_{i=P_x+1}^{P_r} r_i^2\langle\varphi_i^2\rangle}_{\text{Extra terms in true residual}}$$

True residual/PCG



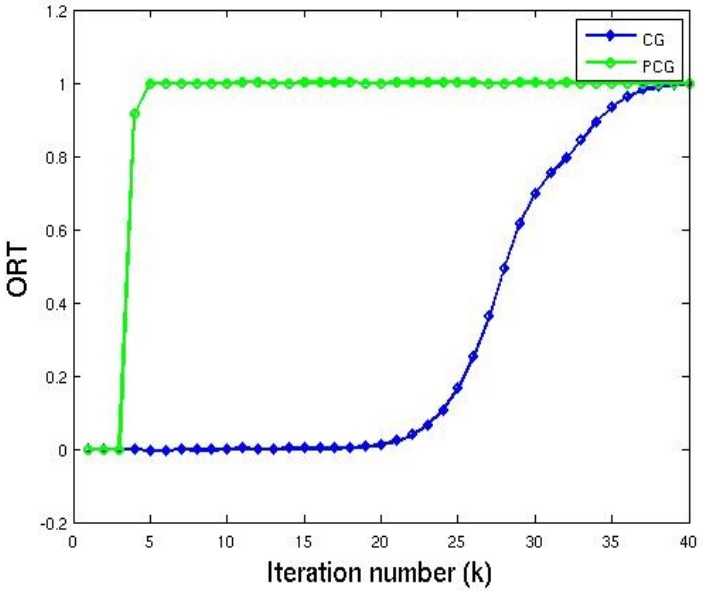
Observations:

- True residual reveals that for k>p_x+1 iterations are inefficient.
- Truncation may impose approximate CG/PCG, consider

$$\left\langle r_k(\xi)^T r_l(\xi)\right\rangle = 0$$

transition exact->Approximate PCG/CG

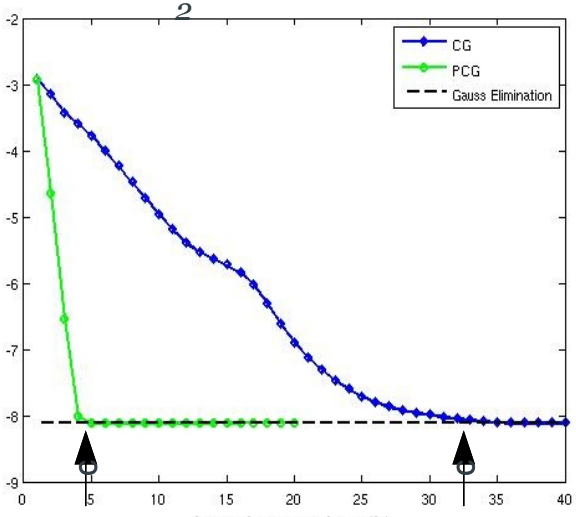
$$ORT = \langle \tilde{r}_{k+1}^T(\xi)\tilde{r}_k(\xi)\rangle / ||\tilde{r}_{k+1}||_2 ||\tilde{r}_k||_2$$



Check

$$|ORT| \leq ORT_{max}$$

L_2 error norm



UNIVERSITY OF Southampton
School of Engineering Sciences

Cost effective stopping criteria

$$\langle \tilde{r}_l^T(\xi)\tilde{r}_m(\xi)\rangle = \underbrace{\sum_{i=0}^{P_x} r_{l,i}^T r_{m,i}\langle\varphi_i^2\rangle}_{o(p_x)} + \delta,$$

$$\delta = \langle [b - A(\xi)x_l(\xi) - r_l(\xi)]^T [b - A(\xi)x_m(\xi) - r_m(\xi)]\rangle$$

δ computed with sparse quadrature, corrects for the terms higher than $o(p_x)$

$$ORT = \langle \tilde{r}_{k+1}^T(\xi)\tilde{r}_k(\xi)\rangle / ||\tilde{r}_{k+1}||_2 ||\tilde{r}_k||_2$$

Additional Computational complexity

1 Low level sparse quadrature

2 PC-scalar products

Memory requirement

PC-variables: x_k, r_k, x_{k+1}, r_{k+1}, w and p ($6 \times P_x \times n$)

UNIVERSITY OF Southampton
School of Engineering Sciences

Large problem within minutes (PCG, Fortran90)

p_x	k_{max}	$\|\|\tilde{r}\|\|_2$	time / (s)
2	3	1.2E-5	4
3	4	2.5E-6	17
4	5	5.6E-7	85
5	6	1.4E-7	366

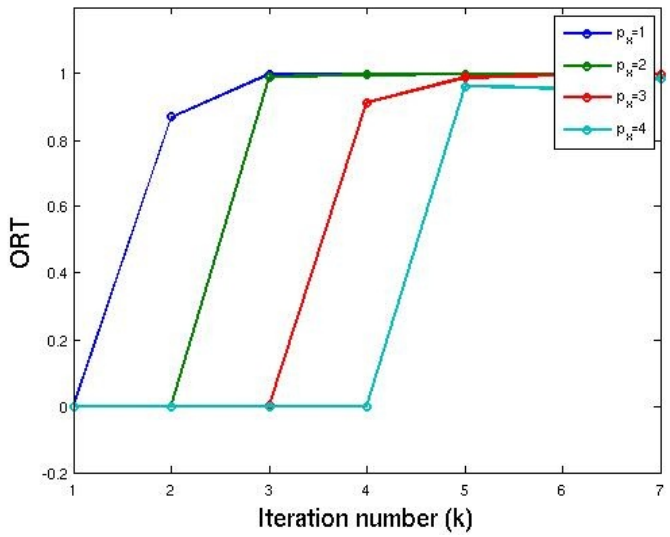
M=10, σ=0.1
mesh size 42000
3003 terms (p_x=5)
126 million unknowns

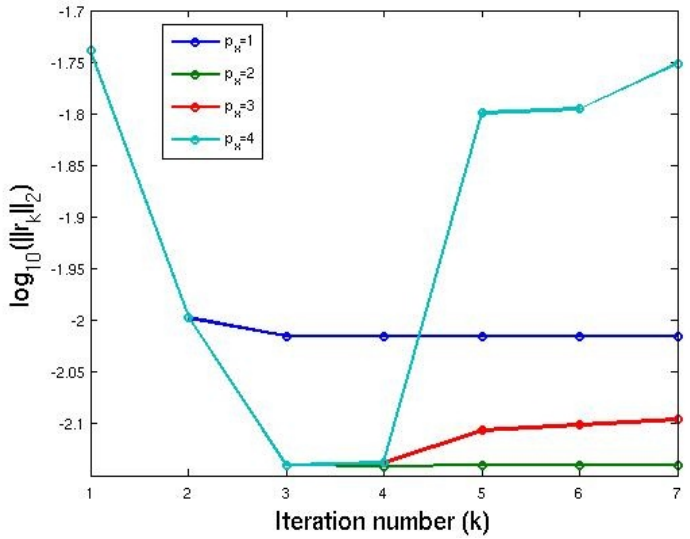
UNIVERSITY OF Southampton
School of Engineering Sciences

Stochastic Galerkin equation may or may not provide SPD A_{big}

σ=0.38, M=4

$$(p_x, \lambda_{min}) \ :(0, \bar{0}.101), (1, 0.08), (2, 0.06), (3, 0.05), (4, 0.007)$$





If iterations are not terminated at the right stage,
then the solution may get corrupted due to inexact CG steps

UNIVERSITY OF Southampton
School of Engineering Sciences

Summary

- PCG algorithm for parametrized random a algebraic equations Matlab,Fortan90
- Efficiency gain by terminating iterations at the exact->approximate transition

UNIVERSITY OF Southampton
School of Engineering Sciences

Future outlook

- Formulation of Krylov methods that minimize alternative error norms (residual and mean square norms)
- Consequence of PC truncation in stochastic GMRES[†]
- Parallel PC library (FORTRAN90, MATLAB) will be freely available with solvers and relevant test matrix generation routines (have done $\langle \varphi_i \varphi_j \varphi_k \rangle$ for M=30 and p=5), p.hakansson@soton.ac.uk

Acknowledgements: Funded by **EPSRC grant EP/F006802/1**

[†]) i.e. non symmetric matrix, for instance convection diffusion problem

UNIVERSITY OF Southampton
School of Engineering Sciences

Divider page title goes here.