FERUM
(Finite Element Reliability Using Matlab)

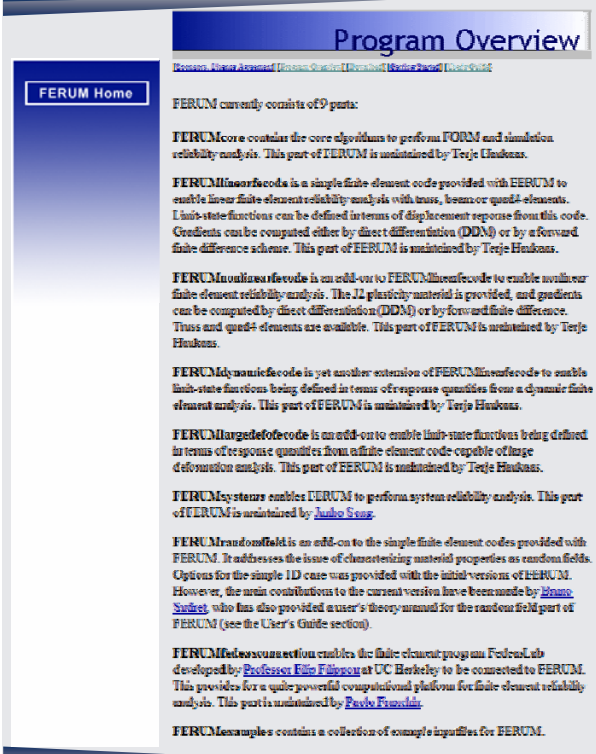
J.-M. Bourinet

LaMI / IFMA - Clermont-Ferrand, France

Un bref historique

- **Création**
  - Eté 1999, UCB – Créateur : T. Haukass
  - Objectif pédagogique à la demande d'A. Der Kiureghian
- **Développements à UC Berkeley**
  - 1999 – 2003 (UCB) – Coordinateur : T. Haukass
  - Implémentation des méthodes fiabilistes usuelles (FORM, SORM, MC/IS, sensibilités) plus quelques unes plus spécifiques (fiabilité système, champs aléatoires, EFS, FORM inverse)
  - Contributions via thésards, stagiaires, post-docs à UC Berkeley
  - Dernière version distribuée : v 3.1 – http://www.ce.berkeley.edu/FERUM/
- **Développements parallèles …**
  - 2001-2002 (UCB, DGA/CTSN) et depuis 2004 (IFMA) – J.-M. Bourinet
  - Reprise de la "philosophie" du code, adaptation de sa structure (vectorisation des calculs, communication avec codes externes), ajout de nouvelles fonctionnalités (sensibilités, DS, subset simulation, ^{2}SMART, RBDO)

Page d'accueil UCB



Principes de base

Structuration des données d'entrée

- Variables de type structure ▶ stockées dans un script **inputfile_xxx.m**

probdata	Données du modèle probabiliste (Nataf)
analysisopt	Options des méthodes de calcul
gfundata	Définition de la (des) fonction(s) de performance
femodel	Paramètres spécifiques au programme appelé par la fonction de performance
randomfield	Définition du champ aléatoire et de ses paramètres
system	Définitions de la représentation minimale par coupes

Lancement de FERUM

```
>> inputfile_xxx
>> ferum
>> (numéro de l'option de calcul)
```

Post-traitement

- Une variable de type structure spécifique à la méthode de calcul employée est créée dans l'environnement de travail. Exemple : **formresults**

Communication avec codes externes

Fonction de performance gfun.m vectorisée (IFMA)

- Tous les algorithmes de calcul fiabiliste sont récrits pour tirer bénéfice de calculs indépendants envoyés et exécutés en simultané
- Autorise un calcul Matlab vectorisé

$$R \sim N\left(7, \frac{5}{\sqrt{3}}\right) \quad S \sim N\left(2, \frac{2}{\sqrt{3}}\right) \quad g = r - s \quad \text{MC : } N = 1.5 \times 10^9$$

Non vectorisé ▶ 6 jours 15 h **Vectorisé ▶ 31 min**

- Permet un calcul distribué sur machine multi-processeurs

Couplages réalisés avec (IFMA) :

- Tous codes avec fichiers d'entrée paramétrés
  (Gibi/gmsh + Aster, Nasgro, XFEM LaMCoS, tous codes EF commerciaux)
- Autres codes avec génération de fichiers via Matlab
  (MAN/EVE LMA Marseille/LaMI)

Structure analysisopt

```
analysisopt.multi_proc = 1;
analysisopt.block_size = 1000;
```

Structure gfundata (cas analytique)

```
% Type of limit-state function evaluator (Alternatives: 'basic', 'aster', 'fcgxfem', 'eve' )
gfundata(1).evaluator = 'basic';
gfundata(1).type = 'expression';

gfundata(1).expression = 'x1.^thetag1 - thetag2*x2 + thetag3';

gfundata(1).dgdq = { ' thetag1*x1.^(thetag1-1) '
                     ' -thetag2 ' };

gfundata(1).thetagname = {'thetag1', 'thetag2', 'thetag3' };
gfundata(1).thetag = [1 1 0];

gfundata(1).flag_sens = 1;
```


Structure gfundata (cas fonction Matlab)

```
% Type of limit-state function evaluator (Alternatives: 'basic', 'aster', 'fcgxfem', 'eve' )
gfundata(1).evaluator = 'basic';
gfundata(1).type = 'expression';
gfundata(1).expression = 'N_Virkler(logC,m,W,e,sigma,R,a0,af,na)-N0';

gfundata(1).cgname = { 'N0' 'W' 'e' 'sigma' 'R' 'af' 'na' };
gfundata(1).cg = [ 400000 6*25.4 0.1*25.4 23.4e3/(6*25.4*0.1*25.4) 0.2 49.8 1001 ];

gfundata(1).flag_sens = 0;
```

Structure gfundata (cas code externe)

```
% Type of limit-state function evaluator (Alternatives: 'basic', 'aster', 'fcgxfem', 'eve' )
gfundata(1).evaluator = 'aster';
gfundata(1).type = 'expression';

gfundata(1).expression  = 'u0 - gext';

gfundata(1).flag_sens = 0;
```


Structure femodel (couplage Gibi + Aster)

```
femodel.data = [ 0    1  nan ;
                 0    2  nan ;
                 0    3  nan ;
                 0    4  nan ;
                 0    5  nan ;
                 0    6  nan ;
                 2    7  nan ;
                 0    8  nan ;
                 1    9  nan ;
                 1   10  nan ;
                 2   11  nan ];

femodel.auxva = 1;

% Computer platform parameters - Enter paths with slashes (not backslashes), even for windows
femodel.mesh_generator      = 'dgib';
femodel.mesh_generator_arg  = '';
femodel.jobdir_name         = 'exec';
femodel.job_name            = 'tuyau';
femodel.keep_optional_files = 0; % 1: keep all files, 0: erase files
```

Structure femodel (couplage Aster sous Windows)

```
if strcmp(analysisopt.client_OS,'win')
   femodel.code_pathname    = 'C:/aster';
   femodel.script_path      = 'C:/temp/aster/seq2/scripts';
   femodel.template_path    = 'C:/temp/aster/seq2/templates';
   femodel.work_parent      = 'C:/temp/aster/seq2/execdir';
   femodel.client_dir       = 'C:/temp/aster/temp2';
   femodel.gawk_path_name   = 'C:/gawk/gawk.exe';
```


Structure femodel (couplage Aster sous Linux)

```
else
   femodel.code_pathname    = '/opt/aster';
   femodel.script_path      = '/home/prof/bourinet/aster/distrib2/scripts';
   femodel.template_path    = '/home/prof/bourinet/aster/distrib2/templates';
   femodel.work_parent      = '/home/prof/bourinet/aster/distrib2/execdir';
   femodel.queue_name       = 'q8h';
   femodel.user_IP          = 'bourinet@172.16.40.1';
   if strcmp(analysisopt.client_OS,'linux')
      femodel.client_dir    = '/home/prof/bourinet/aster/temp2';
   elseif strcmp(analysisopt.client_OS,'master')
      femodel.client_dir    = '/home/prof/bourinet/aster/temp2';
   elseif strcmp(analysisopt.client_OS,'winftp')
      femodel.client_dir    = 'C:/temp/aster/temp2';
      femodel.puttydir      = 'D:/Putty';
      femodel.ftppasswd     = '******';
      femodel.putty_profile = 'bourinet';
   elseif strcmp(analysisopt.client_OS,'winsamba')
      femodel.client_dir    = 'C:/temp/aster/temp2';
      femodel.work_client   = 'Z:/aster/distrib2/execdir';
      femodel.puttydir      = 'D:/Putty';
      femodel.putty_profile = 'bourinet';
   end
end
```

Transformation isoprobabiliste

Modèle de Nataf (UCB & IFMA)

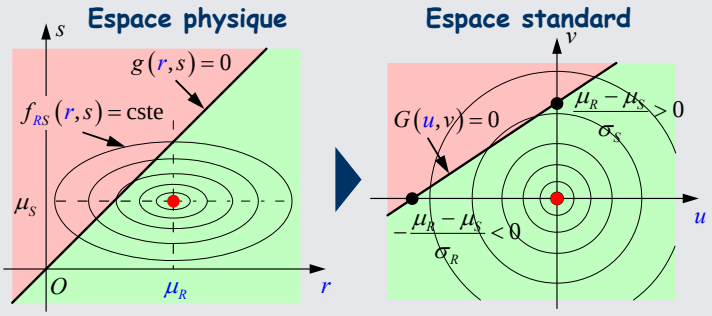
Liu & Der Kiureghian, 1986

$\underline{X}$ avec f_{X_i} , F_{X_i}, $\underline{\underline{R}} = \left[\rho_{ij}\right]_{n\times n}$ $\Rightarrow$ $\underline{Z} \sim N\left(\underline{0}, \underline{\underline{R}}_0\right)$ $\Rightarrow$ $\underline{U} \sim N\left(\underline{0}, \underline{\underline{I}}\right)$

Passage à des lois normales corrélées $\Phi\left(z_i\right) = F_{X_i}\left(x_i\right)$ $\underline{u} = \underline{\underline{L}}_0^{-1}\,\underline{z}$ **Décorrélation**

Matrice de corrélation $\underline{\underline{R}}_0$ calculée :

- de manière analytique par formules approchées (UCB)
- par intégration numérique (IFMA)



Lois disponibles

- Normale
- Normale tronquée (IFMA)
- Lognormale
- Uniforme
- Chi deux
- Exponentielle
- Gamma
- Rayleigh
- Beta
- Gumbel (Type I, E1-max)
- Gumbel (Type I, E1-min)
- Frechet (Type II, E2-max)
- Weibull (Type III, E3-min)

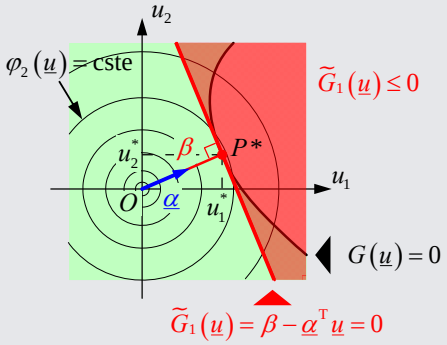
Structure probdata

```
probdata.name =  { 'X1'
                   'X2'
                   'X3'};

probdata.marg =  [  2   500  100   500 nan  nan  nan   nan 0;
                    2  2000  400  2000 nan  nan  nan   nan 0;
                    6     5  0.5     5 nan  nan  nan   nan 0];

probdata.correlation = [ 1.0 0.3 0.2 ;
                         0.3 1.0 -0.2 ;
                         0.2 -0.2 1.0 ];

probdata.transf_type = 3;
probdata.Ro_method = 1;
probdata.flag_sens = 1;
```


First-Order Reliability Method (FORM)

$$G(\underline{u}) \simeq \cancel{G(\underline{u}^*)}^{=0} + \nabla_{\underline{u}} G(\underline{u}^*)^{\mathrm{T}} (\underline{u} - \underline{u}^*) = G_1(\underline{u})$$

$$\underline{u}^* = \underset{\underline{u}}{\operatorname{argmin}} \{ \|\underline{u}\| \mid G(\underline{u}) = 0 \}$$



Algorithme i-HLRF (UCB & IFMA)

Liu & Der Kiureghian, 1991
Zhang & Der Kiureghian, 1997

Gradients évalués par :

- différences finies - Perturbation unique (UCB), choix par variable aléatoire (IFMA)
- différentiation directe - Direct Differentiation Method (DDM)

Possibilité de stockage de tous les paramètres à chaque itération (IFMA)

Structure analysisopt

```
% Maximum number of iterations allowed in the search algorithm
analysisopt.i_max = 500;

% Tolerance on how close design point is to limit-state surface
analysisopt.e1 = 0.001;

% Tolerance on how accurately the gradient points towards the origin
analysisopt.e2 = 0.001;

% 0: step size by Armijo rule, otherwise: given value is the step size.
analysisopt.step_code = 0;

analysisopt.Recorded_u = 1;
analysisopt.Recorded_x = 1;

% 'ddm': direct differentiation, 'ffd': forward finite difference
analysisopt.grad_flag = 'ffd';

% Parameter for computation of FFD gradients - Perturbation = stdv/analysisopt.ffdpara;
% Recommended values: 1000 for basic limit-state functions, 50 for FE-based limit-state functions
analysisopt.ffdpara = 1000;
```


Test

- **Variables aléatoires (2)**

  $X_1 \sim N(0,1)$ $X_2 \sim N(0,1)$

- **Fonction de performance g**

  $g(x_1, x_2) =$ [redacted]

[redacted]

- **Probabilité de défaillance**

  $p_f = 1.83 \times 10^{-3}$ (FORM - 75 appels à g)

  $p_f = 1.95 \times 10^{-3}$ (SORM-cf - 5 appels à g)

  $p_f = 1.97 \times 10^{-3}$ (SORM-pf - 10 appels à g)

  $p_f = 1.93 \times 10^{-3}$ (IS - 1000 appels à g)

Test

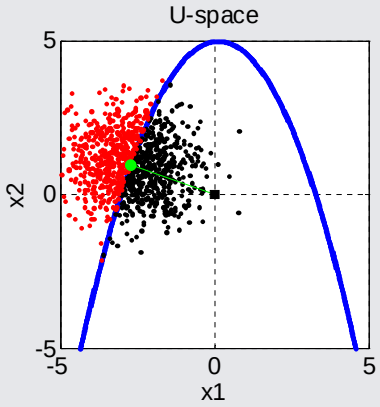
- **Variables aléatoires (2)**

$X_1 \sim N(0,1)$ $\quad X_2 \sim N(0,1)$

- **Fonction de performance g**

$g(x_1, x_2) = b - x_2 - \mu(x_1 - e)^2$

with: $b = 5$, $\mu = 0.5$, $e = 0.1$



- **Probabilité de défaillance**

$p_f = 1.83 \times 10^{-3}$ (FORM - 75 appels à g)

$p_f = 1.95 \times 10^{-3}$ (SORM-cf - 5 appels à g)

$p_f = 1.97 \times 10^{-3}$ (SORM-pf - 10 appels à g)

$p_f = 1.93 \times 10^{-3}$ (IS - 1000 appels à g)

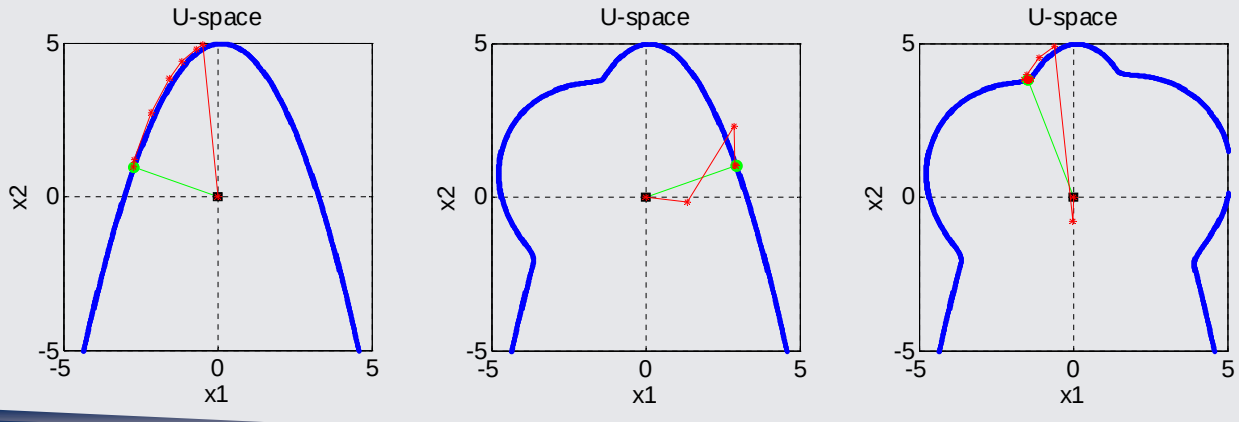
$p_f = 3.016 \times 10^{-3}$ (DS, 100 dir)

$p_f = 3.017 \times 10^{-3}$ (SS, 3×10^5 sim/pas, 500 répl.)

Multi-FORM

- **FORM pour points de conception multiples (IFMA)**

Der Kiureghian & Dakessian, 1998

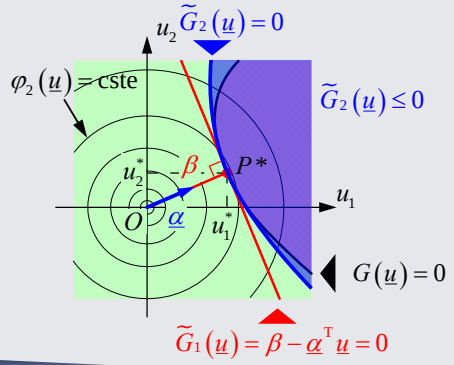


Second-Order Reliability Method (SORM)

$$G(\underline{u}) \simeq \cancel{G(\underline{u}^*)}^{=0} + \underline{\nabla}_u G(\underline{u}^*)^{\mathrm{T}} (\underline{u}-\underline{u}^*) + \frac{1}{2}(\underline{u}-\underline{u}^*)^{\mathrm{T}} \underline{\underline{\nabla}}_u^2 G(\underline{u}^*)(\underline{u}-\underline{u}^*) \quad = G_2(\underline{u})$$

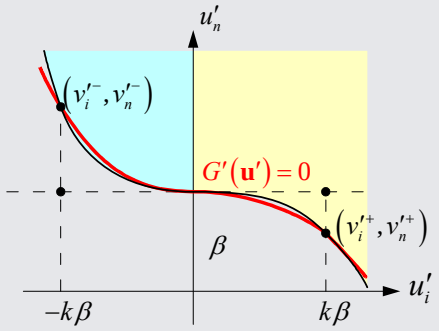
Calcul des courbures en P^* (UCB & IFMA)

$$p_{f2} = \varphi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1+\psi(\beta)\kappa_i}}$$



Lissage par 1/2 paraboloïdes autour de P^* (UCB & IFMA)

Der Kiureghian & De Stefano, 1991



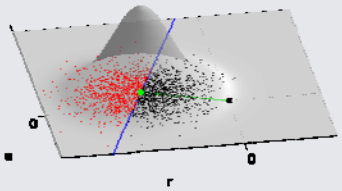
Monte Carlo simulation (MC) / Importance Sampling (IS)

MC (UCB & IFMA)

$$p_f = \int_{D_{f\underline{X}}} I(\underline{x})\, f_{\underline{X}}(\underline{x})\, \mathrm{d}\underline{x} \quad \blacktriangleright \quad p_f = E_{f_{\underline{X}}}\left[I(\underline{X})\right]$$

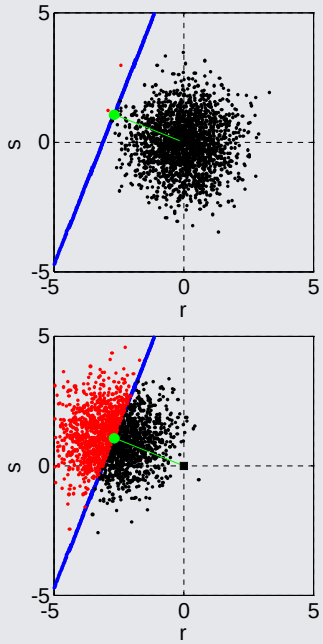
IS (UCB & IFMA)

$$p_f = \int_{D_h} I(\underline{x}) \frac{f_{\underline{X}}(\underline{x})}{h(\underline{x})} h(\underline{x})\, \mathrm{d}\underline{x} \quad \blacktriangleright \quad p_f = E_h\left[I(\underline{X}) \frac{f_{\underline{X}}(\underline{X})}{h(\underline{X})} \right]$$



$$h(\underline{u}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \underline{\underline{\Sigma}}}} \exp\left[-\frac{1}{2}(\underline{u}-\underline{u}^*)^{\mathrm{T}} \underline{\underline{\Sigma}}^{-1} (\underline{u}-\underline{u}^*) \right]$$

avec : $\underline{\underline{\Sigma}} = \underline{\underline{\mathrm{I}}}$



$$\hat{p}_f = \frac{1}{N}\sum_{i=1}^{N} q_i$$

avec $q_i = I(\underline{x}_i)$ (MC) ou $q_i = I(\underline{x}_i)\dfrac{f_{\underline{X}}(\underline{x}_i)}{h(\underline{x}_i)}$ (IS)

Structure analysisopt

```
% 'dspt': design point, 'origin': origin in standard normal space (simulation analysis)
analysisopt.sim_point = 'origin';

% 0: default rand matlab function, 1: Mersenne Twister (preferred)
analysisopt.rand_generator = 1;

% Standard deviation of sampling distribution in simulation analysis
analysisopt.stdv_sim  = 1;

% Number of simulations in simulation analysis
analysisopt.num_sim   = 1000000;

% Target coefficient of variation of failure probability in simulation analysis
analysisopt.target_cov = 0.01;
```

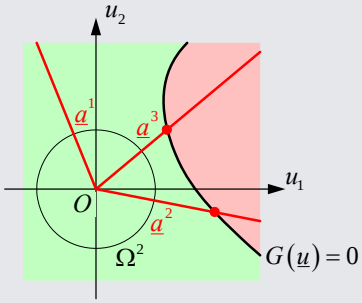

Directional Simulation (DS)

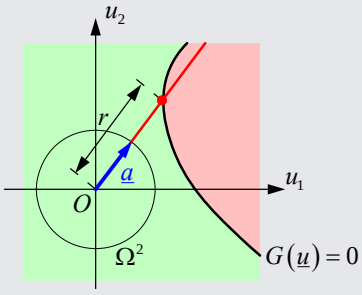
Simulation directionelle (IFMA)

Schueller & Stix, 1987

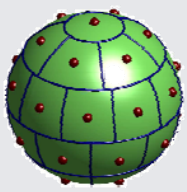
$$p_f = \int_{\underline{a} \in \Omega^n} P\left[\, G(R\underline{A}) \le 0 \mid \underline{A} = \underline{a} \,\right] f_{\underline{A}}(\underline{a})\, \mathrm{d}\underline{a}$$

$$\hat{p}_f = \frac{1}{N}\sum_{i=1}^{N}\left[1 - \chi_n^2\left(r_i^2\right)\right]$$





Directions via points choisis de manière équirépartie sur l'hypersphère de rayon unité (IFMA)



Structure analysisopt

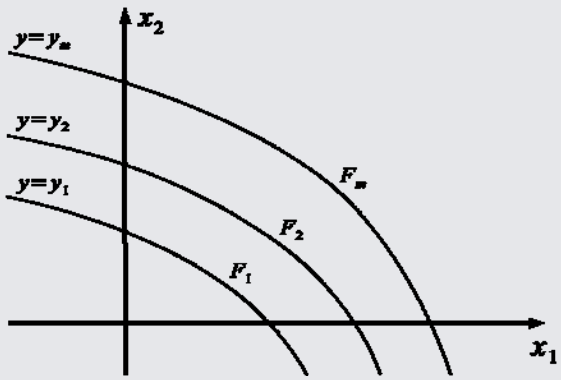
```
% Max search radius in standard normal space for directional simulation analysis
analysisopt.rho = 8;

% Tolerance for searching zeros of g function
analysisopt.tolx = 1e-5;

% Flag for storage of a values which gives axes along which simulations are carried out
analysisopt.keep_a = 1;

% Flag for storage of r values for which g(r) = 0
analysisopt.keep_r = 1;
```

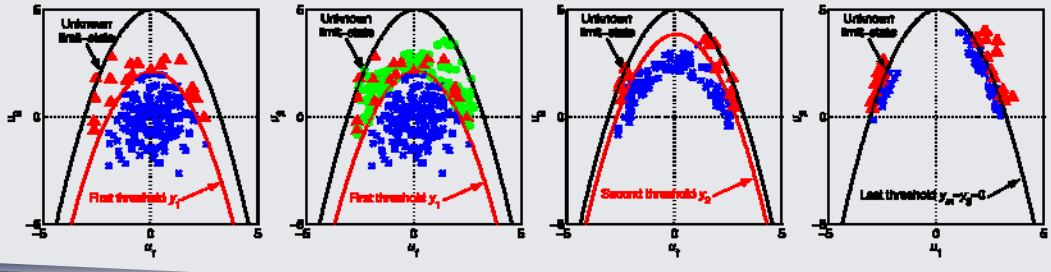

Subset Simulation (SS)

Subset simulation (IFMA)

Au & Beck, 2001

$$
\begin{aligned}
p_f &= P(F) \\
&= P(F \mid F_{m-1}) P(F_{m-1}) \\
&= \dots \\
&= P(F_1) \prod_{i=2}^{m} P(F_i \mid F_{i-1})
\end{aligned}
$$





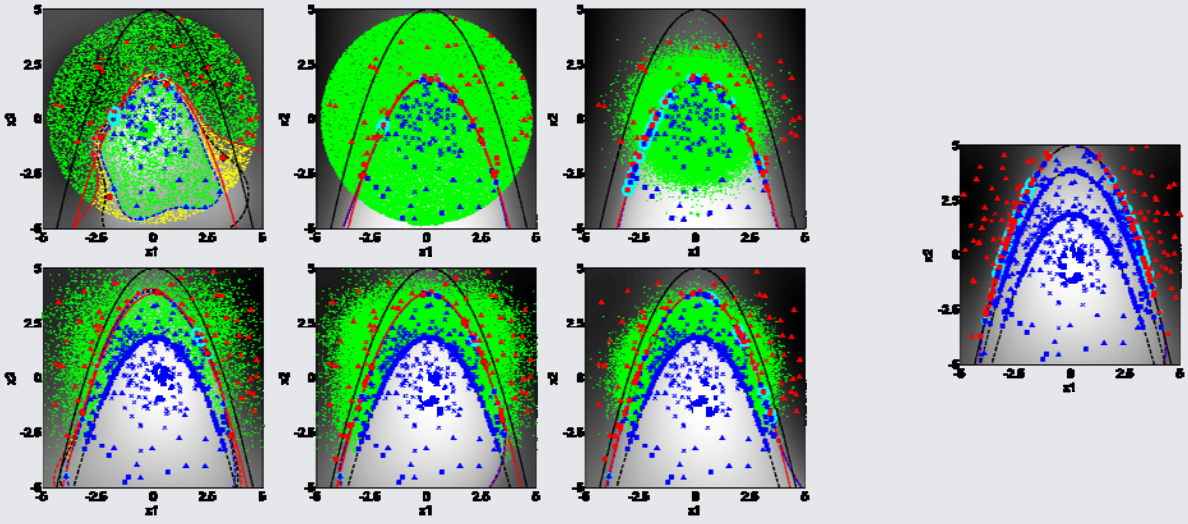
Structure analysisopt (suite)

```
analysisopt.width = 2;

analysisopt.pf_target = 0.1;

% i>=0 : restart from step i, -inf : full no save (default), -1 : full save
analysisopt.ss_restart_from_step = -inf;
```


Subset Simulation and Support Vector Machines (^{2}SMART)

- **Subset simulation et SVMs (IFMA)**

  Deheeger, Bourinet & Lemaire, 2008



Structure analysisopt (suite)

```
analysisopt.num_sim_sdu = [ 50 50 ];

analysisopt.num_sim_norm = [ 100 100 ];

nrv = size(probdata.marg,1);
analysisopt.buf_size = round(1.25e7/nrv);
analysisopt.svm_buf_size = 3500^2;

analysisopt.flag_var_rbf = 0;
```


Facteurs d'importance

Variables non-corrélées

$$\underline{\alpha} = -\frac{\nabla_u G(\underline{u}^*)}{\left\|\nabla_u G(\underline{u}^*)\right\|}$$

α - factors
Cosinus directeurs

Variables corrélées

$$\underline{\gamma}^{\mathrm{T}} = \frac{\underline{\alpha}^{\mathrm{T}} \underline{\underline{J}}_{u,x} \underline{\underline{\hat{D}}}}{\left\|\underline{\alpha}^{\mathrm{T}} \underline{\underline{J}}_{u,x} \underline{\underline{\hat{D}}}\right\|}$$

γ - factors

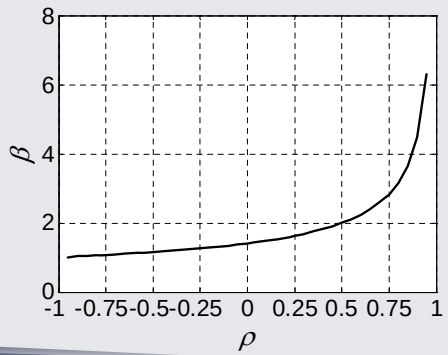
$\underline{\gamma}$ n'est en général pas un vecteur unitaire

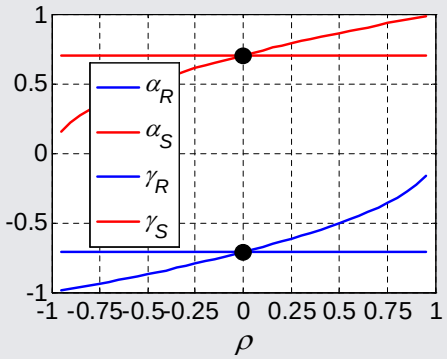
$\underline{\gamma} = \underline{\alpha}$ en l'absence de corrélation

$R \sim N(4,1)$
$S \sim N(2,1)$

$\rho = [-0.95, 0.95]$

$g = r - s$





Sensibilités fiabilistes

Sensibilités fiabilistes

$$p_f = \int_{g(\underline{x},\underline{\theta}_g)\le 0} f_{\underline{X}}\left(\underline{x},\underline{\theta}_f\right) \mathrm{d}\,\underline{x}$$

$\underline{\theta}_f$: paramètres des lois - Ex. : $\mu, \sigma, p_1, p_2, \ldots, \rho$

$\underline{\theta}_g$: paramètres (déterministes) de l'état-limite

- Sensibilités de β par rapport aux paramètres des lois

$$\nabla_{\underline{\theta}_f}\beta = \underline{\alpha}^{\mathrm{T}}\,\underline{\underline{J}}_{T,\underline{\theta}_f}\left(\underline{x}^*,\underline{\theta}_f\right)$$

où T : $\underline{x} \to \underline{u} = T\left(\underline{x},\underline{\theta}_f\right)$

- ✓ Sensibilités de β par rapport aux paramètres de l'état-limite

$$\nabla_{\underline{\theta}_g}\beta = \frac{1}{\left\|\nabla_{\underline{u}} G\left(\underline{u}^*,\underline{\theta}_g\right)\right\|}\nabla_{\underline{\theta}_g} g\left(\underline{x}^*,\underline{\theta}_g\right)$$

- ✓ Sensibilités de p_f par rapport aux paramètres des lois / de l'état-limite

$$\begin{aligned}\nabla_{\underline{\theta}_f \text{ ou } \underline{\theta}_g} p_{f1} &= -\varphi(-\beta)\,\nabla_{\underline{\theta}_f \text{ ou } \underline{\theta}_g}\beta \\ &= -\varphi(\beta)\,\nabla_{\underline{\theta}_f \text{ ou } \underline{\theta}_g}\beta\end{aligned}$$

Sensibilités fiabilistes (suite)

Sensibilités aux paramètres des distributions (UCB & IFMA)

- calcul approché (UCB)
- Calcul "exact" par intégration numérique (IFMA)

$$\frac{\partial \underline{u}}{\partial \theta_f} = \underline{\underline{L}}_0^{-1}\frac{\partial \underline{z}}{\partial \theta_f} + \frac{\partial \underline{\underline{L}}_0^{-1}}{\partial \theta_f}\underline{z}$$

$\theta_f = \mu_i, \sigma_i$

$= p_{1i}, p_{2i}, \ldots$

Classiquement pris en compte — Généralement négligé

Sensibilités à la corrélation entre variables aléatoires (IFMA)

$$\frac{\partial u_k}{\partial \rho_{ij}} = \sum_{l=1}^{k}\frac{\partial l_{0,kl}^{-1}}{\partial \rho_{ij}}\Phi^{-1}\left[F_{X_l}\left(x_l\right)\right]$$

$\theta_f = \rho_{ij}$

Structures probdata, analysisopt et gfundata

```
probdata.flag_sens = 1;
```

$\nabla_{\underline{\theta}_f} \beta \ , \ \nabla_{\underline{\theta}_f} p_{f1}$

```
% Parameter for computation of FFD estimates of dbeta_dthetag
% Perturbation = thetag/analysisopt.ffdpara_thetag if thetag ~= 0
%             or 1/analysisopt.ffdpara_thetag if thetag == 0;
% Recommended values: 1000 for basic limit-state functions,
%                      100 for FE-based limit-state functions
analysisopt.ffdpara_thetag = 1000;

gfundata(1).flag_sens = 0;
```

$\nabla_{\underline{\theta}_g} \beta \ , \ \nabla_{\underline{\theta}_g} p_{f1}$

Sensibilité à la corrélation

Geometry ▶ Center-cracked tension specimen M(T)

$W = 6"$: width $\quad \sigma = 60.45$ MPa : max applied stress

$e = 0.1"$: thickness $\quad R = 0.2$: stress ratio

Limit-state function

$$g(C,m) = N_R(C,m) - N_S$$

where $\quad N_R = \int_{a_i}^{a_f} \frac{1}{C\Delta K^m} \, \mathrm{d}a$

N_S : target service life

with a_i : initial crack half-length

$a_f = 49.8$ mm : final crack half-length

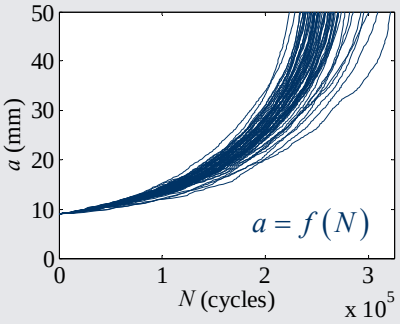
$$\frac{\mathrm{d}a}{\mathrm{d}N} = C\,\Delta K^m \quad \text{(Paris-Erdogan law)}$$

$$\Delta K = K_{\max}(1-R) \quad \text{and} \quad K_{\max} = \frac{1 - 0.025(a/W)^2 + 0.06(a/W)^4}{\sqrt{\cos(\pi a/W)}} \sigma\sqrt{\pi a}$$

Sensibilité à la corrélation

Stochastic model

Statistics of C and m from Virkler's data (68 samples)



Virkler et *al.*, 1979

Case #1

Variable	Type	Mean	Stdv.	Correlation
a_i	–	4.4	–	–
$\ln C$	normal	–26.056	0.972	–0.99795
m	normal	2.855	0.166	

Case #2

Variable	Type	Mean	Stdv.	Correlation
a_i	exp.	1.5	1.5	–
$\ln C$	normal	–26.056	0.972	–0.99795
m	normal	2.855	0.166	

Sensibilité à la corrélation

Case #1

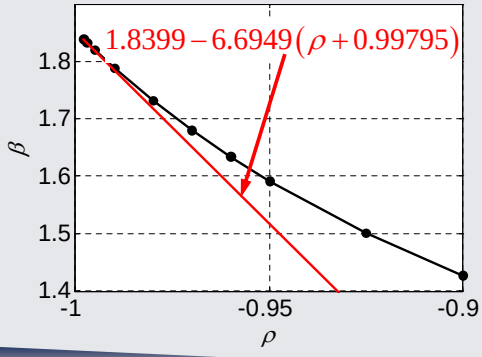
$\beta = 1.8005$ $\quad N_S = 400000$ cycles

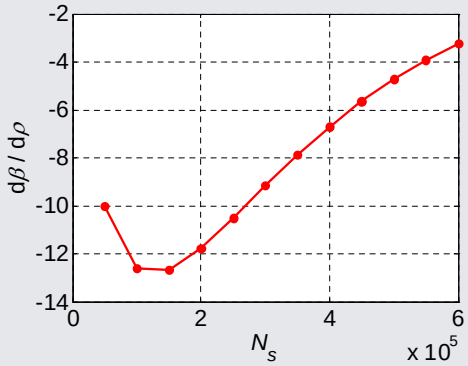
$\dfrac{\mathrm{d}\beta}{\mathrm{d}\rho} = -368.39$

Case #2

$\beta = 1.8399$ $\quad N_S = 400000$ cycles

$\dfrac{\mathrm{d}\beta}{\mathrm{d}\rho} = -6.6949$





Reliability-Based Design Optimization (RBDO – N2LA)

Subset simulation et SVMs (IFMA)

Royset & Polak, 2004 (RBDO basée sur MC and IS)

Royset & Polak, 2007 (RBDO basée sur DS)

Nested bi-Level Approach (N2LA)

- Méthode d'optimisation fiabiliste basée sur gradients
- Méthode fiabiliste appelée à l'intérieur de la boucle d'optimisation
- Algorithme d'optimisation de Polak-He (Polak, 1997)
- Recherche de la direction de descente basée sur l'algorithme SQP
- Pas dans la direction de descente suivant règle d'Armijo

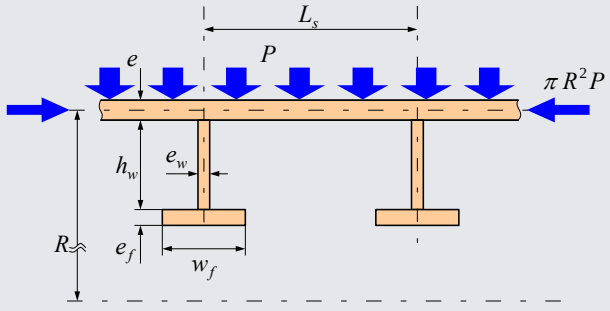
Travaux IFMA

- Résolution de la dépendance aux ordres de grandeur des fonctions coût, contraintes et variables d'optimisation
- Implémentation de FORM, FORM système, MC, IS, DS
- Vectorisation des algorithmes (partiel)

Structures rbdo_fundata et rbdo_parameters

```
rbdo_fundata.cost = {       '4/3.*pi.*(r1.^3 - r0.^3)'
                     '0e2 * 4/3.*pi.*(r1.^3 - r0.^3)' };

rbdo_fundata.constraint = { '- r0 + 40'
                                '- r1'
                            'r0 - 150'
                         'r1 - 200000'
                             'r0 - r1'};

rbdo_parameters.alpha = 0.5;
rbdo_parameters.beta  = 0.6;
rbdo_parameters.gamma = 1;
rbdo_parameters.delta = 1;
rbdo_parameters.tau   = 0.9999;
rbdo_parameters.eta   = 0.0001;

rbdo_parameters.max_iter = 10;
rbdo_parameters.Nmax = 1e8;
rbdo_parameters.steplim = 20;
rbdo_parameters.target_improvement_ratio = 0; % * initial_cost_value
rbdo_parameters.target_beta = 4.5;
% Underlying reliability method:'FORM' or 'FORM_Serials' or 'MC' or 'dirsim'
rbdo_parameters.method = 'FORM';
```

Exemple RBDO – Collapse plastique d'une coque raidie



Variables aléatoires

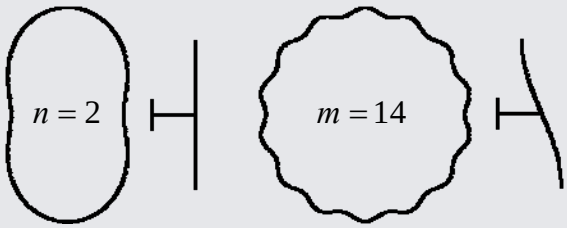
Variable	Distribution	Moyenne	C.d.v.
A_{n2}	Lognormal		50%
A_{m14}	Lognormal		50%
E	Lognormal	200 GPa	3%
σ_y	Lognormal	390 MPa	5%

Variables d'optimisation déterministes

R	2488	h_w	156
L_s	600	e_w	10
e	24	w_f	120
		e_f	24

Imperfections prise en compte

$$A(\theta,z) = A_n \cos n\theta + A_m \cos\frac{\pi z}{L_s}\cos m\theta$$



Fonction coût

M (poids de la coques) / V (déplacement)

Contraintes déterministes

Critères de déversement de raidisseur (BS 5500)

Exemple RBDO

```
rbdo_fundata.cost =
{       '7850 * ( ((2488+e/2)^2-(2488-e/2)^2)*600 + ((2488-e/2)^2-(2488-
    (e/2+hw))^2)*ew + ((2488-(e/2+hw))^2-(2488-(e/2+hw+ef))^2)*wf ) / ( 1026 *
    (2488+e/2)^2 * 600 )'

  '0e2 * 7850 * ( ((2488+e/2)^2-(2488-e/2)^2)*600 + ((2488-e/2)^2-(2488-
    (e/2+hw))^2)*ew + ((2488-(e/2+hw))^2-(2488-(e/2+hw+ef))^2)*wf ) / ( 1026 *
    (2488+e/2)^2 * 600 )' };

rbdo_fundata.constraint = { ' -e + 10 '
                            ' -hw + 75 '
                            ' -ew + 5'
                            ' -wf + 50 '
                            ' wf – 175 '
                            ' -ef + 5 '
                            ' hw - (1.1*(2e5/390)^0.5) * ew '
                            ' wf/2 - (0.5*(2e5/390)^0.5) * ef '
                            ' hw - 2*wf '
                            ' wf - hw ' };
```

Distribution de FERUM

Planning prévisionnel de mise en ligne

Fin 2008

A faire :

- Rédaction d'un guide utilisateur, d'un recueil d'exemples
- Site web (IFMA et miroir UCB)
- … un peu d'ordre à apporter et des commentaires à ajouter

Philosophie

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007
http://www.gnu.org/licenses/gpl.html

Vous êtes libre d'utiliser … mais vos contributions sont les bienvenues ☺

OpenSees

Site internet

http://opensees.berkeley.edu/

Document de référence (partie fiabiliste)

http://www.inrisk.ubc.ca/opensees/
ubc_opensees_guide.pdf