Contents


```
%-------------------------- Question 6 ---------------------------------
```

Saltelli estimators of Sobol indices

```
This code illustrates the implementation of the Monte Carlo estimators
for computing the first first-order indices and total effects indices
   for ishigami function
y=sin(x1) + a*(sin(x2))^2 + b*(x3^4)*sin(x1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc
```

Setup the model and define input ranges

number of parameters and parameter ranges

```
p = 3;
param1 =  [-pi  pi];
```

Sample parameter space:

number of samples

```
M = 10000;
```

Compute [A], [B] matrices and [C] as random variables

Using random samples from the uniform distributions A(:,1) = param1(1) + (param1(2) - param1(1)).*rand(M,1); A(:,2) = param2(1) + (param2(2) - param2(1)).*rand(M,1); A(:,3) = param3(1) + (param3(2) - param3(1)).*rand(M,1);

B(:,1) = param1(1) + (param1(2) - param1(1)).*rand(M,1); B(:,2) = param2(1) + (param2(2) - param2(1)).*rand(M,1); B(:,3) = param3(1) + (param3(2) - param3(1)).*rand(M,1);

```
% Using Latin hypercube samples (LHS) from the uniform distributions
% This approach converges with smaller M compared to random samples
% since LHS spreads the samples more evenly across the parameters space
A_lhs = lhsdesign(M,p);
B_lhs = lhsdesign(M,p);
params = param1;
A = zeros(size(A_lhs));
B = zeros(size(B_lhs));
for i = 1:p
    A(:,i) = params(2) - (params(2) - params(1)).*A_lhs(:,i);
    B(:,i) = params(2) - (params(2) - params(1)).*B_lhs(:,i);
end
```

Compute [C] matrices

```
C = zeros(M,p,p);
for i = 1:p
    C(:,:,i) = B;
    C(:,i,i) = A(:,i);
end
```

Run the model and compute selected model output at sampled parameter

```
for  j = 1:M
    yA(j,1) = ishigami(A(j,:),7,0.1);
    yB(j,1) = ishigami(B(j,:),7,0.1);
    for i = 1:p
        yC(j,i) = ishigami(C(j,:,i),7,0.1);
    end
end
```

Compute sensitivity indices

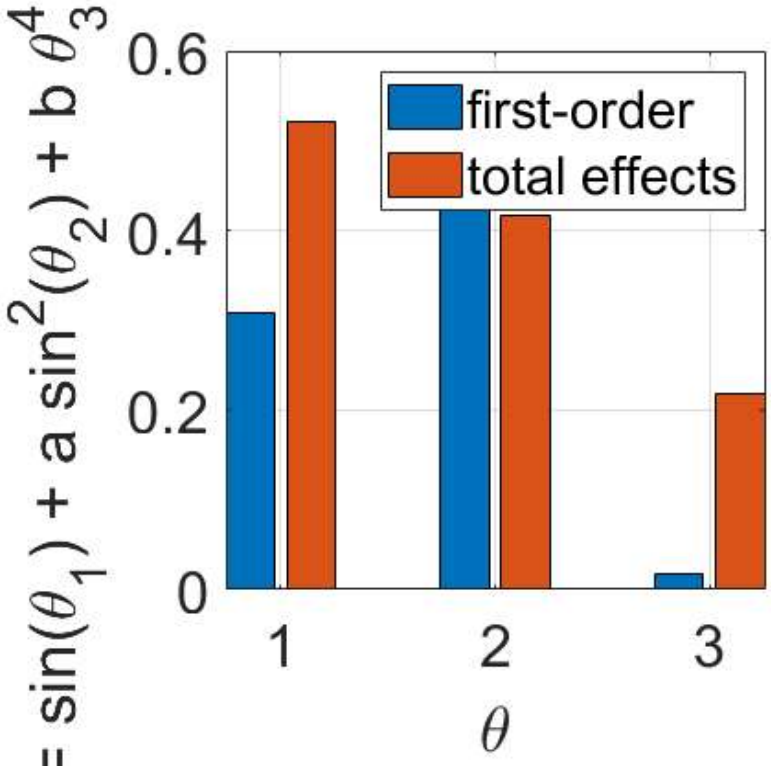
```
f0  = mean(yA) ;
VARy = mean(yA.^2) - f0^2 ;

for i = 1:p
    yCi = yC(:,i);

        % fist order indices
    Si(i)  = ( 1/M*sum(yA.*yCi) - f0^2 ) / VARy ;
    % total effects indices
    STi(i) = 1 -  ( 1/M*sum(yB.*yCi) - f0^2 ) / VARy ;
end
```

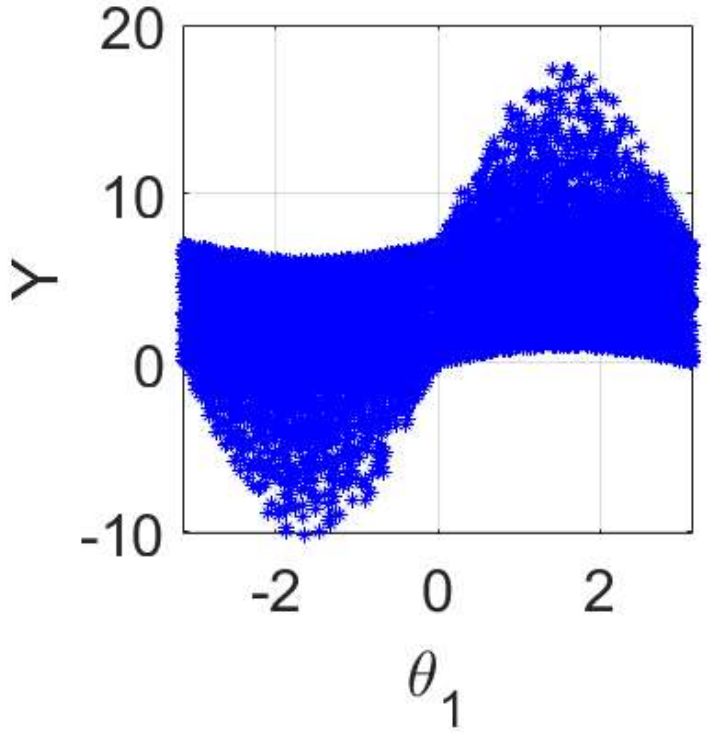
Plot results

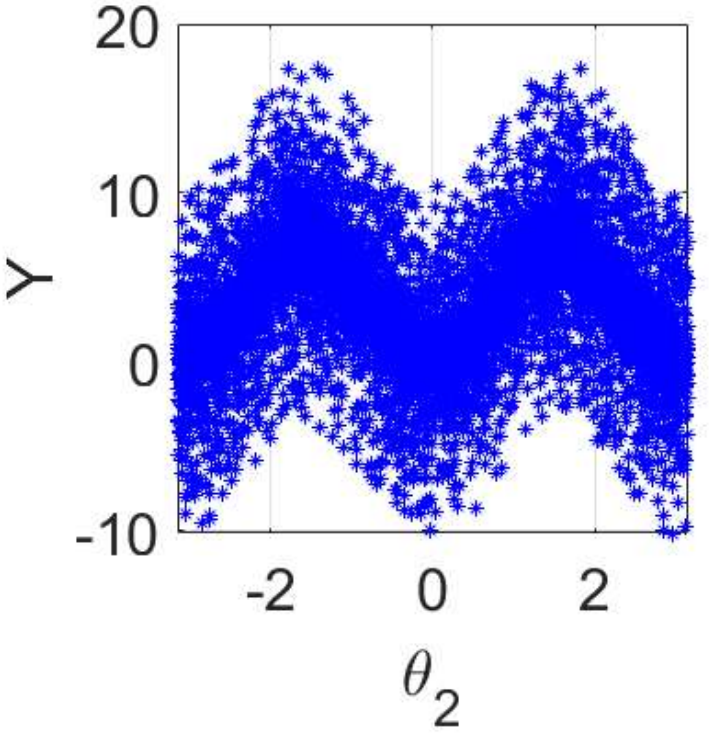
sensitivity indices

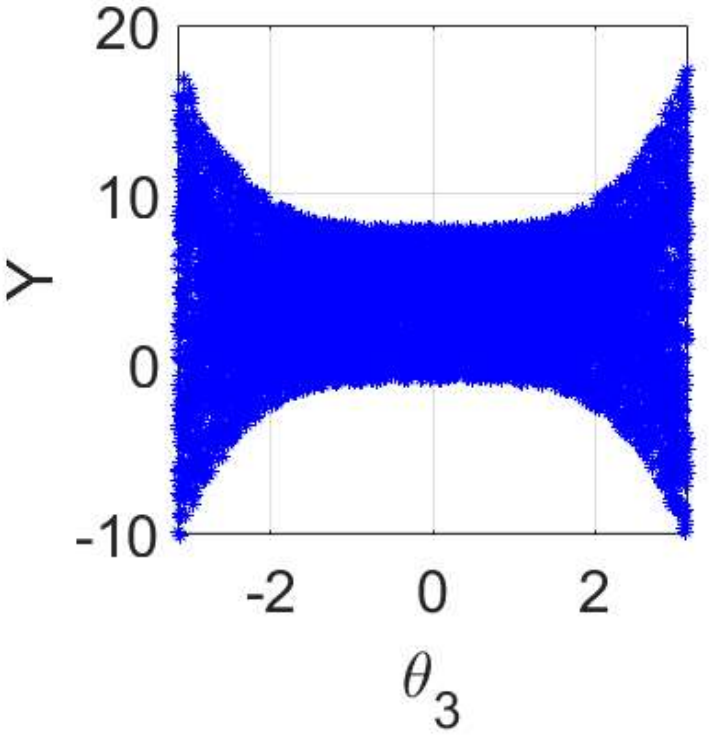
```
indices = [Si' STi'];

fprintf('Si Indices are :')
indices(:,1)

fprintf('STi Indices are :')
indices(:,2)

figure
bar(indices)
axis square,xlabel('\theta'),ylabel('Y = sin(\theta_1) + a sin^2(\theta_2) + b \theta_3^4 sin(\theta_3)'), grid on
set(gca,'FontSize',24)
legend('first-order', 'total effects')

% scatter plots
figure
plot(A(:,1), yA, '*b')
axis square,xlabel('\theta_1'),ylabel('Y'), grid on
set(gca,'FontSize',24)

figure
plot(A(:,2), yA, '*b')
axis square,xlabel('\theta_2'),ylabel('Y'), grid on
set(gca,'FontSize',24)

figure
plot(A(:,3), yA, '*b')
axis square,xlabel('\theta_3'),ylabel('Y'), grid on
set(gca,'FontSize',24)
```

```
Si Indices are :
ans =

    0.3092
    0.4504
    0.0167

STi Indices are :
ans =

    0.5202
    0.4177
    0.2185
```


= sin(θ1) + a sin2(θ2) + b θ34
0.6
0.4
0.2
0
first-order
total effects
1
2
3
θ


20
10
0
-10
Y
-2
0
2
θ1





Published with MATLAB® R2020a