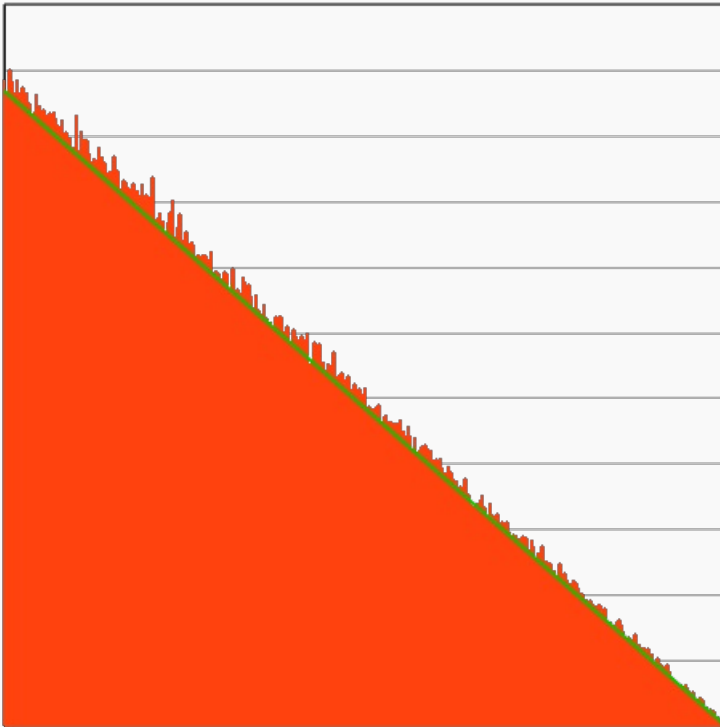


Simulation de la loi de probabilités de l'écart absolu de deux nombres réels pris aléatoirement entre 0 et 1 (problème du duel du magasin "Tangente").

1. La largeur des intervalles se paramètre par le nombre de décimales (exemple, decimal\_largeur=2 correspond à une largeur de 0,01).
2. Les fréquences observées sont les aires des rectangles rouges (et non les hauteurs de ces rectangles) afin de montrer le lien entre les probabilités et les aires par calcul intégral.
3. La hauteur de l'affichage graphique s'adapte en fonction des paramètres.
4. La fonction de densité la plus cohérente s'affiche en commentant ou en décommentant la dernière instruction (//). Cette technique ne fonctionne pas forcément en «mode normal», elle fonctionne en «mode édition».
5. Au niveau de la sortie graphique, l'axe des abscisses représente «valeur simulée de l'écart» et l'axe des ordonnées «densité des valeurs simulées».

### Graphique



Xmin: 0 ; Xmax: 1  
 Ymin: 0 ; Ymax: 2.2715  
 GradX: 0 ; GradY: 0.2065

### Code de l'algorithme

```

1  VARIABLES
2  n1 EST_DU_TYPE NOMBRE
3  n2 EST_DU_TYPE NOMBRE
4  ecart EST_DU_TYPE NOMBRE
5  compteur1 EST_DU_TYPE NOMBRE
6  compteur2 EST_DU_TYPE NOMBRE
7  compteur3 EST_DU_TYPE NOMBRE
8  decimal_largeur EST_DU_TYPE NOMBRE
9  nb_tirage EST_DU_TYPE NOMBRE
10 j_graphique EST_DU_TYPE NOMBRE
11 hauteur EST_DU_TYPE NOMBRE
12 statistique EST_DU_TYPE LISTE
13 DEBUT_ALGORITHME
14 LIRE decimal_largeur
15 LIRE nb_tirage
16 POUR compteur1 ALLANT_DE 0 A pow(10,decimal_largeur)-1
17   DEBUT_POUR
18   statistique[compteur1] PREND_LA_VALEUR 0
19   FIN_POUR
20 POUR compteur2 ALLANT_DE 1 A nb_tirage
21   DEBUT_POUR
22   n1 PREND_LA_VALEUR random()
23   n2 PREND_LA_VALEUR random()
24   ecart PREND_LA_VALEUR abs(n1-n2)
25   ecart PREND_LA_VALEUR pow(10,decimal_largeur)*ecart
26   ecart PREND_LA_VALEUR floor(ecart)
27   statistique[ecart] PREND_LA_VALEUR statistique[ecart]+1
28   FIN_POUR
29 // Début de l'obtention de la hauteur maximale des rectangles (paramétrages graphique).
30 hauteur PREND_LA_VALEUR
    
```

```

(ALGOBOX_MAXIMUM(statistique,0,pow(10,decimal_largeur)-1)/nb_tirage)*pow(10,decimal_largeur)
31 // Fin de l'obtention de la hauteur maximale des rectangles (paramétrages graphique).
32 POUR compteur3 ALLANT_DE 0 A pow(10,decimal_largeur)-1
33   DEBUT_POUR
34     statistique[compteur3] PREND_LA_VALEUR statistique[compteur3]/nb_tirage
35     j_graphique PREND_LA_VALEUR pow(10,decimal_largeur)*statistique[compteur3]
36     TRACER_SEGMENT (compteur3/pow(10,decimal_largeur),0)->((compteur3+1)/pow(10,decimal_largeur),0)
37     TRACER_SEGMENT (compteur3/pow(10,decimal_largeur),j_graphique)->
((compteur3+1)/pow(10,decimal_largeur),j_graphique)
38     TRACER_SEGMENT (compteur3/pow(10,decimal_largeur),0)->(compteur3/pow(10,decimal_largeur),j_graphique)
39     TRACER_SEGMENT ((compteur3+1)/pow(10,decimal_largeur),0)->
((compteur3+1)/pow(10,decimal_largeur),j_graphique)
40   FIN_POUR
41 // Décommentez l'instruction ci-dessous afin de tracer une éventuelle fonction de densité.
42 TRACER_SEGMENT (0,2)->(1,0)
43 FIN_ALGORITHME

```

## Résultats

```

***Algorithme lancé***
Entrer decimal_largeur : 3
Entrer nb_tirage : 1000000
***Algorithme terminé***

```