

Rapport Hebdo

Viet Anh Quach

3SR

4 septembre 2025

Changer le modèle

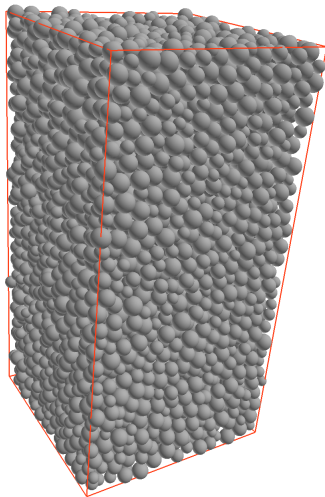


Figure 1 – Boîte rectangulaire

Étude sur le nombre d'inertie

Symboles	Paramètres	Valeurs	Unité
Nombre de particules	N	$15 \times 30 \times 15 = 6750$	-
Le rayon des particules	R	$0.003 \div 0.005$	m
Masse volumique	ρ	2500	kg/m^3
Raideur normale et tangentielle	k_n & k_t	3×10^6	N/m
Niveau de raideur	κ	>1000	-
Coefficient de frottement	μ	$\mu_{iso} = 0.1, \mu_{triax} = 0.5$	-
Coefficient d'amortissement	α	0.0	-

Table 1 – Valeurs gardé

Étude sur le nombre d'inertie

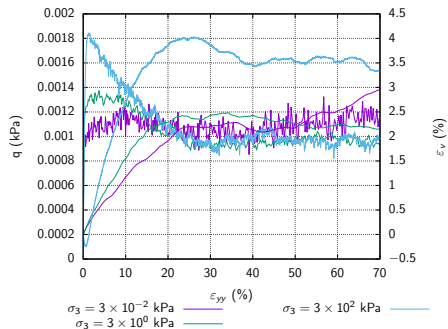
$v(\text{m/s})$	$\sigma_3 = 3 \times 10^2 \text{ (kPa)}$
4.542×10^{-3}	$I = 10^{-5}$
4.542×10^{-2}	$I = 10^{-4}$
4.542×10^{-1}	$I = 10^{-3}$
4.542×10^0	$I = 10^{-2}$
4.542×10^1	$I = 10^{-1}$
4.542×10^2	$I = 1$

Table 2 – Changer la vitesse

$\sigma_3(\text{kPa})$	$v = 4.542 \times 10^{-2} \text{ (m/s)}$
3×10^4	$I = 10^{-5}$
3×10^2	$I = 10^{-4}$
3×10^0	$I = 10^{-3}$
3×10^{-2}	$I = 10^{-2}$
3×10^{-4}	$I = 10^{-1}$
3×10^{-7}	$I = 1$

Table 3 – Changer la contrainte de confinement

Changer la contrainte

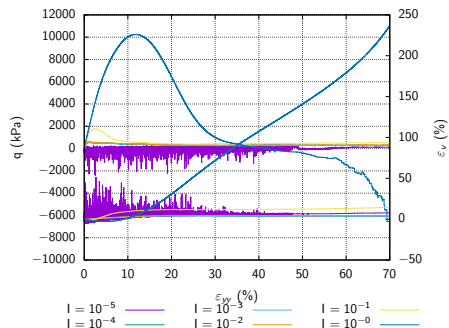


σ_3 (kPa)	$v = 4.542 \times 10^{-2}$ (m/s)
3×10^4	$\kappa > 1000$
3×10^2	$l = 10^{-4}$
3×10^0	$\tan \varphi \approx 90^\circ$
3×10^{-2}	
4.542×10^1	IsoComp stabilise pas
4.542×10^2	

Table 4 – Changer la contrainte de confinement

Figure 2 – Contrainte - Déformation DEM (changer la vitesse)

Changer la vitess



$v(\text{m/s})$	$\sigma_3 = 3 \times 10^2 \text{ (kPa)}$
4.542×10^{-3}	$l = 10^{-5}$
4.542×10^{-2}	$l = 10^{-4}$
4.542×10^{-1}	$l = 10^{-3}$
4.542×10^0	$l = 10^{-2}$
4.542×10^1	$l = 10^{-1}$
4.542×10^2	$l = 1$

Table 5 – Changer la vitess

Figure 3 – Contrainte - Déformation DEM (changer la vitess)

Changer la vitesse

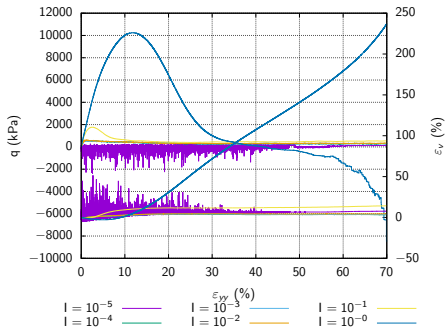


Figure 4 – Contrainte - Déformation DEM (changer la vitesse)

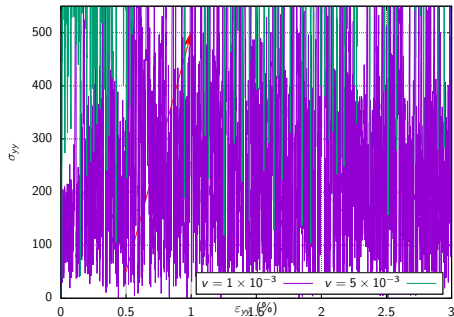


Figure 5 – Bruyant concernant pas de temps MPM (avant)

$$\dot{x}(t) = \frac{x(t + \varepsilon) - x(t - \varepsilon)}{2\varepsilon}$$

Problème de arrondir ?

Comparer entre les formes de boîte

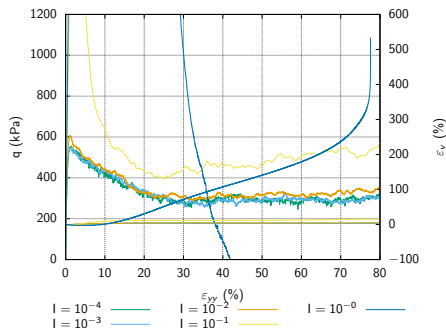


Figure 6 – Rectangulaire

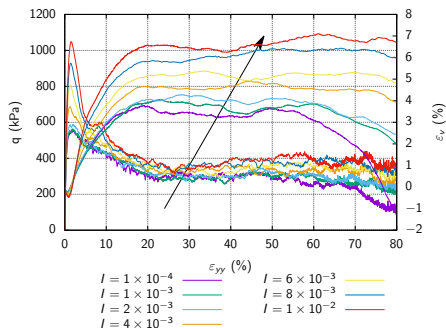


Figure 7 – Cube (précédent)

En compression quasi-statique, la contrainte déviatorique au pic ou à l'état critique (donc μ) ne présente aucune différence entre les deux formes.

Comparer entre les formes de boîte

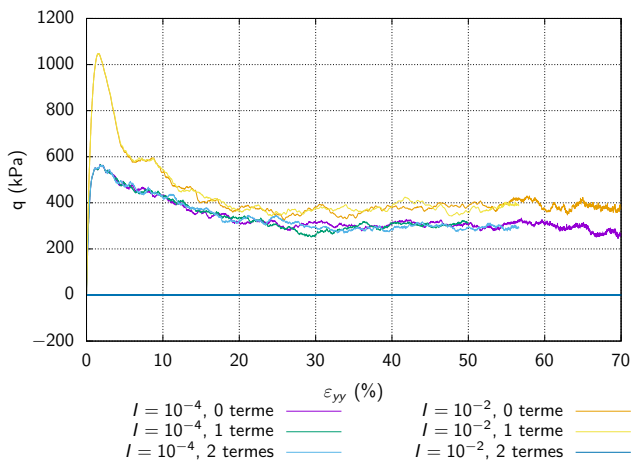


Figure 8 – Courbe Contrainte ($\sigma_3 = 300\text{kPa}$)

Problème d'arrondir (Standard IEEE 754)

- Un type flottant ne représente qu'un nombre limité de chiffres significatifs ; au-delà, la valeur devient inexacte.
- Manipulation délicate à cause des différences entre binaire et décimal.
- Les opérations mathématiques amplifient les erreurs d'arrondi (e.g : $+$ et \times).

Problème d'arrondir (Standard IEEE 754)

- La comparaison de valeurs flottantes peut poser problème.
- Si les opérandes sont très proches, les opérateurs (surtout `=` et `!=`) deviennent peu fiables.

```

Test.cpp > @main()
1 #include <iomanip>
2 #include <iostream>
3 #include <algorithm>
4
5 int main()
6 {
7     std::cout << std::setprecision(17);
8
9     constexpr double d1{1.0};
10    constexpr double d2{0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1};
11    std::cout << std::boolalpha << (d2 == 1.0) << '\n'; // true
12
13    constexpr double d3{100.0 - 99.99}; // 0.01
14    constexpr double d4{10.0 - 9.99}; // 0.01
15
16    if (d3 == d4)
17        std::cout << "d3 == d4" << '\n';
18    else if (d3 > d4)
19        std::cout << "d3 > d4" << '\n';
20    else if (d3 < d4)
21        std::cout << "d3 < d4" << '\n';
22
23    std::cout << std::boolalpha << (0.3 == 0.2 + 0.1) << '\n'; // true
24
25    return 0;
26 }

```

PROBLEME OUTPUT DEBUG CONSOLE TERMINAL PORTS SPLIT CHECKER ADVICE

```

vquach@GT-GP0050:~/Documents/rapportThese$ cd "/home/35-LAB/vquach/Documents/rapportThese/" && g++ -std=c++11 -g -c *.cpp
false
d3 > d4
false
vquach@GT-GP0050:~/Documents/rapportThese$

```

Problème d'arrondir (Standard IEEE 754)

- The Art of Computer Programming, Volume II : Seminumerical Algorithms (Addison-Wesley, 1969)"
- Si elles sont « assez proches » à ϵ , on les considère comme égales.

```

1 #include <algorithm> // for std::max
2 #include <cmath>     // for std::abs
3 #include <iostream>
4
5 // Return true if the difference between a and b is within epsilon percent of the larger of a and b
6 bool approximatelyEqualRel(double a, double b, double relEpsilon)
7 {
8     return (std::abs(a - b) <= (std::max(std::abs(a), std::abs(b)) * relEpsilon));
9 }
10
11 // Return true if the difference between a and b is less than or equal to absEpsilon, or within relEpsilon percent of the
12 bool approximatelyEqualAbsRel(double a, double b, double absEpsilon, double relEpsilon)
13 {
14     // Check if the numbers are really close -- needed when comparing numbers near zero.
15     if (std::abs(a - b) <= absEpsilon)
16         return true;
17     // Otherwise fall back to Knuth's algorithm
18     return approximatelyEqualRel(a, b, relEpsilon);
19 }
20
21 int main()
22 {
23     // a is really close to 1.0, but has rounding errors
24     constexpr double a(0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1);
25
26     constexpr double relEps(1e-8);
27     constexpr double absEps(1e-12);
28
29     std::cout << std::boolalpha; // print true or false instead of 1 or 0
30
31     std::cout << approximatelyEqualRel(a, 1.0, relEps) << '\n'; // compare "almost 1.0" to 1.0
32     std::cout << approximatelyEqualRel(a - 1.0, 0.0, relEps) << '\n'; // compare "almost 0.0" to 0.0
33
34     std::cout << approximatelyEqualAbsRel(a, 1.0, absEps, relEps) << '\n'; // compare "almost 1.0" to 1.0
35     std::cout << approximatelyEqualAbsRel(a - 1.0, 0.0, absEps, relEps) << '\n'; // compare "almost 0.0" to 0.0
36
37     return 0;
38 }

```

PROBLEME OUTPUT SERIAL CONSOLE TERMINAL PORTS SPELL CHECKER ASURE

```

+ vquach@GT-QW050:~/Documents/rapportThese$ cd "/home/3S-LAB/vquach/Documents/rapportThese/" && g++ -std=c++23 roundingError.cpp
true
false
true
true
vquach@GT-QW050:~/Documents/rapportThese$

```