An elementary model of nuclear fission reactor using the collision probability method

Construction and use

Pierre ALBARÈDE
http://pierre.albarede.free.fr

toolbox

1. Toolbox

1.1. List manipulation

Make sure the list has only one element and take it:

In[1]:=
```
Clear @ only ; only @ {x_} := x
```

In[ ]:=
```
only @ {1}
only @ Range @ 3
```

Out[ ]= 1

Out[ ]= only[{1, 2, 3}]

Create indexed rule list from vector:

In[2]:=
```
Clear @ threadRule ;
threadRule @ r_ := Thread [Table [rᵢ, {i, Length @ #}] → #] &
```

In[ ]:=
```
threadRule [R]@{a, b, c}
```

Out[ ]= {R_1 → a, R_2 → b, R_3 → c}

Inversely:

In[ ]:=
```
Last /@ %
```

Out[ ]= {a, b, c}

In[4]:=
```
Attributes @ hasNegativeElement = {HoldFirst };
Clear @ hasNegativeElement ;
hasNegativeElement @ list1_List := SelectFirst [Unevaluated @ list1 , Negative , {}] =!= {}
```

Exits as soon as possible:

In[ ]:=
```
hasNegativeElement @ {1, Print @ hello ; 2}
```

hello

Out[ ]= False

```
In[ ]:= hasNegativeElement @{-1, Print @hello ; 2}
Out[ ]= True
```

Strictly Upper Triangular matrix reduction:

```
SUT = MapIndexed [Drop[#1, only @#2] &, Most @#] &;

Clear @SUTMatrixForm ;
SUTMatrixForm @SUTMatrix : _List .. :=
 MatrixForm @MapIndexed [Join[Table[□, only @#2], #1] &, Append [SUTMatrix , {}]]
```

```
In[ ]:= With[{matrix1 = Table[10 * i + j, {i, 4}, {j, 4}]},
  {MatrixForm @matrix1 , MatrixForm @SUT @matrix1 , SUTMatrixForm @SUT @matrix1 }]
```

$$\left\{\begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{pmatrix}, \begin{pmatrix} \{12, 13, 14\} \\ \{23, 24\} \\ \{34\} \end{pmatrix}, \begin{pmatrix} □ & 12 & 13 & 14 \\ □ & □ & 23 & 24 \\ □ & □ & □ & 34 \\ □ & □ & □ & □ \end{pmatrix}\right\}$$

Linearly rescale list between given values:

```
In[ ]:= With[{l1 = {10, 12, 14, 16}, min = 5, max = 8},
  With[{min1 = Min @#, max1 = Max @#},
    Composition [min + (# - min1)/(max1 - min1) (max - min) &(*,Log[10,#]&*)] /@ #
   ] &@ l1
 ]
Out[ ]= {5, 6, 7, 8}
```

```
In[9]:= Clear @rescale ;
rescale [l1_List , {min_ , max_}] := With[{min1 = Min @#, max1 = Max @#},
    Composition [min + (# - min1)/(max1 - min1) (max - min) &(*,Log[10,#]&*)] /@ #
   ] &@ l1
```

```
In[ ]:= rescale [{10, 12, 14, 16}, {5, 8}]
Out[ ]= {5, 6, 7, 8}

In[ ]:= rescale [#, {5, 8}] &@ Map[Log[10, #] &, 1. {3, 30, 300, 3000}]
Out[ ]= {5., 6., 7., 8.}
```

With log scale:

In[10]:=
```
Clear @ rescaleLog ;
rescaleLog [l1_List , {min_ , max_}](*/;Select [l1 ,Negative ]=={}*) :=
 rescale [#, {min , max}] &@ Map[Log[10 , #] &, l1]
```

In[ ]:=
```
rescaleLog [{.1 , 1. , 10 , 100 , 1000}, {0, 1}]
```

Out[ ]=
```
{0. , 0.25 , 0.5 , 0.75 , 1.}
```

1.2. Formula manipulation

In[11]:=
```
Clear @ expandSum ;
expandSum [expr_ , n_] := expr /. HoldPattern @ Sum[x_ , j_] :> Sum[x, {j , n}]
```

```
expandSum [∑_j k_j R_j , n]
```

Out[ ]=
$$\sum_{j}^{n} k_j R_j$$

```
expandSum [∑_j k_j R_j , 3]
```

Out[ ]=
$$k_1 R_1 + k_2 R_2 + k_3 R_3$$

```
expandSum [∑_i k_i R_i , 3]
```

Out[ ]=
$$k_1 R_1 + k_2 R_2 + k_3 R_3$$

In[12]:=
```
Clear @ exchange ;
exchange [i_ , j_] := With [{uk = Unique @ k}, # /. {i → uk , j → i} /. uk → j] &
```

In[ ]:=
```
exchange[i, j][∑_j k_j R_j]
exchange[i, j]@P_ij
exchange[j, i]@P_ij
exchange[i, k]@P_ij
exchange[i, 2]@P_ij
exchange[3, j]@P_ij
```

Out[ ]= $\sum_i k_i R_i$

Out[ ]= $P_{j\,i}$

Out[ ]= $P_{j\,i}$

Out[ ]= $P_{k\,j}$

Out[ ]= $P_{2\,j}$

Out[ ]= $P_{i\,3}$

```
simplify::what = "Simplifying  `1` `2` second ";
simplify::stopped = "Simplify  failed  time  constraint ";
Clear@simplify;
simplify[0, _] := Identity;
simplify[time_, label_] := (Message[simplify::what, label, time];
    TimeConstrained[Simplify@#, time, Message@simplify::stopped;
     #]) &;
```

Factor monomial in rational fraction:

In[19]:=
```
Clear@monomialCollect;
monomialCollect@variables_List :=
 Divide @@
   Through[Map[Composition[Total@MonomialList[#, Flatten@variables] &, Expand, #] &,
      {Numerator, Denominator}]@#] &
```

```
In[ ]:= With[
         {rf1 = (1 - k3 P33 + k2 (-k3 P23 P32 + P22 (-1 + k3 P33))) /
              (k1 k3 P13 ((1 - k2 P22) P31 + k2 P21 P32) + k2 k3 P23 (P32 + k1 (P12 P31 - P11 P32)) +
                 (1 - k2 P22 + k1 (-k2 P12 P21 + P11 (-1 + k2 P22))) (-1 + k3 P33))},
         With[{mcrf1 = monomialCollect [Table[ki, {i, 3}]]@rf1},
          simplify[1, "rational  fraction "][rf1 == mcrf1]
         ]
        ]
```

simplify : Simplifying rational fraction 1 seconds

```
Out[ ]= True
```

```
In[ ]:= monomialCollect [{k2, k3}][(0.42 + k2 (-0.40 + 0.38 k3) - 0.40 k3) / (0.85 - 0.93 k3 + k2 (-0.93 + 1.` k3))]
```

$$\text{Out[ ]=}\ \frac{0.42 - 0.4\, k_2 - 0.4\, k_3 + 0.38\, k_2\, k_3}{0.85 - 0.93\, k_2 - 0.93\, k_3 + 1.\, k_2\, k_3}$$

tetrahedron

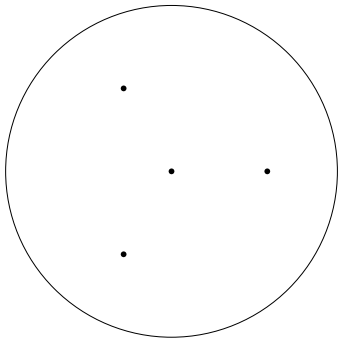
1.3. Regular tetrahedron

```
In[286]:= tetrahedronData  = PolyhedronData ["Tetrahedron ", "VertexCoordinates "]
```

$$\text{Out[286]=}\ \left\{\left\{0, 0, \sqrt{\frac{2}{3}} - \frac{1}{2\sqrt{6}}\right\}, \left\{-\frac{1}{2\sqrt{3}}, -\frac{1}{2}, -\frac{1}{2\sqrt{6}}\right\}, \left\{-\frac{1}{2\sqrt{3}}, \frac{1}{2}, -\frac{1}{2\sqrt{6}}\right\}, \left\{\frac{1}{\sqrt{3}}, 0, -\frac{1}{2\sqrt{6}}\right\}\right\}$$

```
In[287]:= Graphics [{Circle [{0, 0}, 1], Point @* Most /@ tetrahedronData }, Axes → False ,
           ImageSize → Small ]
```

Out[287]=

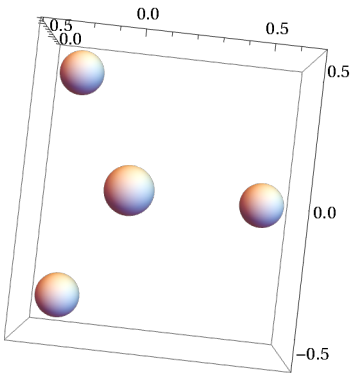


```
In[288]:= EuclideanDistance  @@ # & /@ Subsets [tetrahedronData  , {2}]
```

```
Out[288]= {1, 1, 1, 1, 1, 1}
```

In[289]:=
```
Graphics3D [{Sphere [#, .1] & /@ tetrahedronData (*,{Opacity @.5,Sphere [{0,0,0},1]}*)},
 Axes → True, ImageSize → Small, ViewPoint → {0, 0, 5}]
```

Out[289]=



1.4. Numerical functions

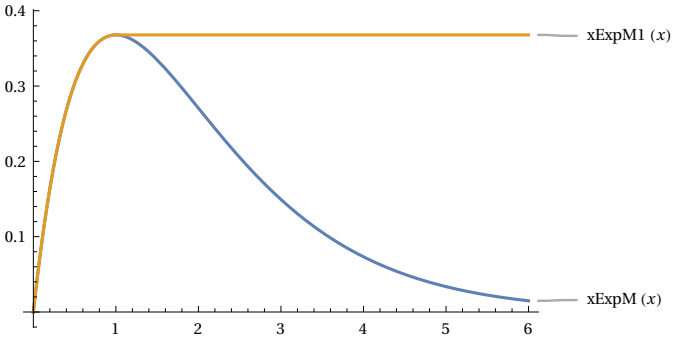
```
Clear @ expM ; expM @ x_ := e^-x /; NumberQ @ x
Clear @ xExpM ; xExpM @ x_ := x e^-x /; NumberQ @ x
Clear @ xExpM1 ;
xExpM1 @ x_ := xExpM @ Min[x, 1] /; NumberQ @ x
```

In[297]:=
```
xExpM @ x == x expM @ x /. x → .123
```

Out[297]=
```
True
```

In[298]:=
```
Plot[{xExpM @ x, xExpM1 @ x}, {x, 0, 6}, PlotLabels → Automatic ]
```

Out[298]=

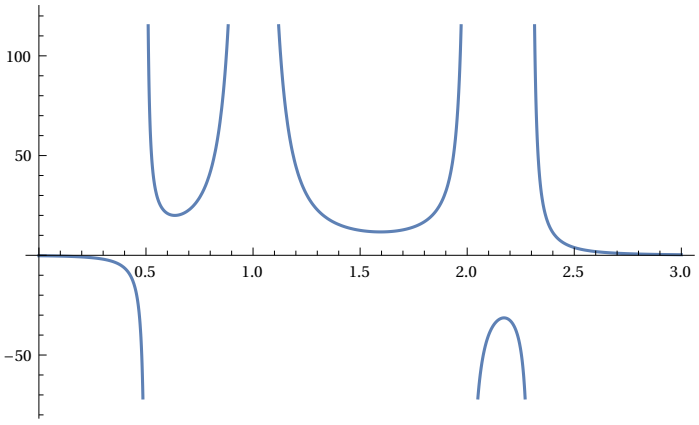


Plot a polynomial fraction, taking only some branches:

```
Attributes @ plotBranch = (*like Plot :*){HoldAll };
Clear @ plotBranch ;
plotBranch [expr_ , {var_ , min_ , max_}, branch_ ,
  (*convention : use option for sequence , options for list ,
  the plural s meaning list of*)
  option : OptionsPattern []] :=
 Plot[expr ,
  {var , Sequence @@ Through [{First , Last}@#] &@@
    Flatten @
     Take[
      Partition [
       {min , Sequence @@ Select [var /. Union @ NSolve [Denominator @ expr == 0, var],
           min < # < max &], max}, 2, 1], branch ]}, option ]
```

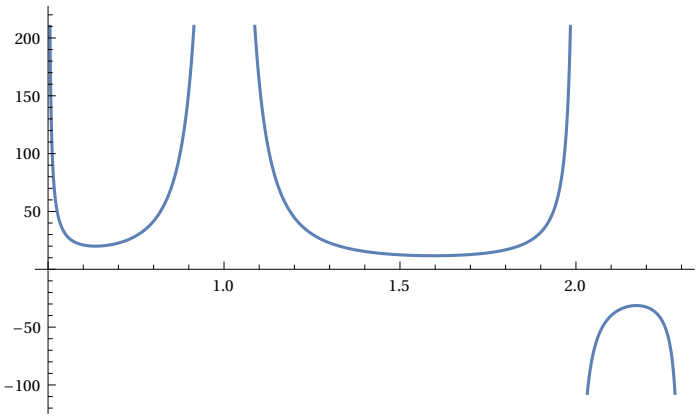
In[ ]:= $\text{plotBranch}\left[\frac{x+1}{(2x-1)(x-1)^2(x-2)(x-2.3)}, \{x, 0, 3\}, \text{All}\right]$

Out[ ]=



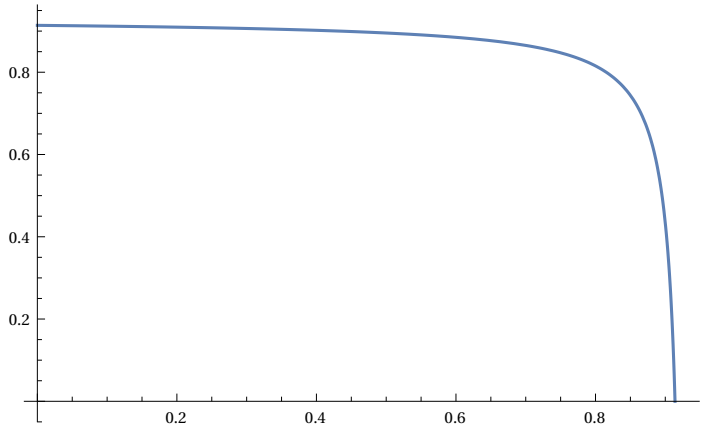
In[ ]:= $\text{plotBranch}\left[\frac{x+1}{(2x-1)(x-1)^2(x-2)(x-2.3)}, \{x, 0, 3\}, \{2, 4\}\right]$

Out[ ]=



In[ ]:=
```
plotBranch[(-0.85 + 0.93 k2)/(-0.93 + k2), {k2, 0, 1.2`}, 1, AxesOrigin → {0, 0}]
```

Out[ ]=



Numerical error as may be used by FindRoot inferred from the Mathematica documentation of FindRoot and AccuracyGoal:

In[26]:=
```
Options @ almostEqual = {
   AccuracyGoal → WorkingPrecision / 2, PrecisionGoal → WorkingPrecision / 2,
   WorkingPrecision -> MachinePrecision , MachinePrecision → $MachinePrecision };
Clear @ almostEqual ;
almostEqual [x_, y_, option___ : OptionsPattern []] :=
 Abs[x - y] < (10 ^ - AccuracyGoal ) + Abs @ x 10 ^ - PrecisionGoal /. {option} //.
  Options @ almostEqual
```

In[ ]:=
```
$MachinePrecision
```

Out[ ]=
```
15.9546
```

In[ ]:=
```
almostEqual [100 , 100.0000001 , PrecisionGoal → Infinity ]
almostEqual [100 , 100.00000001 , PrecisionGoal → Infinity ]
```

Out[ ]=
```
False
```

Out[ ]=
```
True
```

In[ ]:=
```
almostEqual [100 , 100.00001 , AccuracyGoal → Infinity ]
almostEqual [100 , 100.000001 , AccuracyGoal → Infinity ]
```

Out[ ]=
```
False
```

Out[ ]=
```
True
```

In[ ]:=
```
almostEqual [100 , 100.00001 ]
almostEqual [100 , 100.000001 ]
```

Out[ ]=
```
False
```

Out[ ]=
```
True
```

1.5. Numerical equation solving

What is really the FindRoot error function?

There is no symmetry between AccuracyGoal and PrecisionGoal even though Abs@x==1:

```
In[104]:= x /. FindRoot[x^2 - 1.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 1,
            PrecisionGoal → Infinity] // NumberForm[#, {17, 16}] &
Out[104]//NumberForm= 1.1237244333863730

In[105]:= x /. FindRoot[x^2 - 1.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → Infinity,
            PrecisionGoal → 1] // NumberForm[#, {17, 16}] &
FindRoot : The root has been bracketed as closely as possible with machine precision but the function value
    exceeds the absolute tolerance 0.
Out[105]//NumberForm= 1.1234567890123460
```

Ignoring (omitting or setting to Infinity) PrecisionGoal yields a simple behaviour:

```
In[113]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 0,
            PrecisionGoal → Infinity] // NumberForm[#, {17, 16}] &
Out[113]//NumberForm= 100.1234726613452000

In[114]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 0] //
           NumberForm[#, {17, 16}] &
Out[114]//NumberForm= 100.1234726613452000

In[ ]:= x /. FindRoot[x^2 - 1.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 1,
            PrecisionGoal → Infinity] // NumberForm[#, {17, 16}] &
Out[ ]//NumberForm= 1.1237244333863730

In[115]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 4,
            PrecisionGoal → Infinity] // NumberForm[#, {17, 16}] &
Out[115]//NumberForm= 100.1234567883280000

In[116]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 4] //
           NumberForm[#, {17, 16}] &
Out[116]//NumberForm= 100.1234567883280000

In[117]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 15,
            PrecisionGoal → Infinity] // NumberForm[#, {17, 16}] &
Out[117]//NumberForm= 100.1234567890124000

In[118]:= x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, AccuracyGoal → 15] //
           NumberForm[#, {17, 16}] &
Out[118]//NumberForm= 100.1234567890124000
```

In[119]:=
```
x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}, PrecisionGoal → Infinity] //
 NumberForm[#, {17, 16}] &
```

Out[119]//NumberForm=
```
100.1234567890124000
```

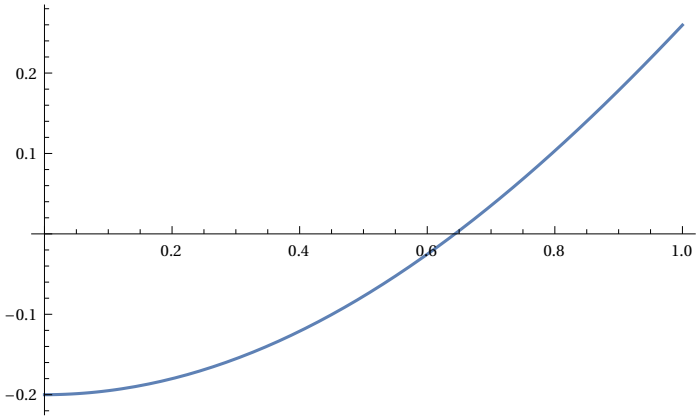
In[120]:=
```
x /. FindRoot[x^2 - 100.12345678901234567^2, {x, .5, 1.5}] // NumberForm[#, {17, 16}] &
```

Out[120]//NumberForm=
```
100.1234567890124000
```

Bisection

In[ ]:=
```
Plot[-Cos@x + .8, {x, 0, 1}]
```

Out[ ]=



Alternative to FindRoot for tests.

In[28]:=
```
Attributes@bisection = {HoldAll};
bisection::nochangeofsign = "sign does not change, try other bounds";
Clear@bisection;
bisection[expr_, {x_, x1_, x2_},
  OptionsPattern[{AccuracyGoal → $MachinePrecision/2, sow → False}]] :=
 (If[OptionValue@sow, Print@only@Last@#];
    (*like FindRoot:*){x → First@#}) &@@
  Reap@
   With[
    {sign = Which[LessEqual @@ #, 1, GreaterEqual @@ #, -1, True,
          Message@bisection::nochangeofsign] &@@
       {ReleaseHold[Hold@expr /. x → x1], 0, ReleaseHold[Hold@expr /. x → x2]}},
    Mean@With[{hook = If[OptionValue@sow, Sow, Identity]},
      NestWhile[
       If[sign ReleaseHold[Hold@expr /. x → hook@#2] > 0, {First@#, #2},
           {#2, Last@#}] &[#, Total@#/2] &, {x1, x2},
       Abs[First@# - Last@#] > Power[10, -OptionValue@AccuracyGoal] &
       (*Not@almostEqual[First@#,Last@#,option]&*)]]]
```

In[299]:=
```
bisection[-Cos@x + x, {x, 0, 1.5}, sow → True]
```

```
{0.75 , 0.375 , 0.5625 , 0.65625 , 0.703125 , 0.726563 , 0.738281 , 0.744141 , 0.741211 , 0.739746 ,
 0.739014 , 0.73938 , 0.739197 , 0.739105 , 0.739059 , 0.739082 , 0.739094 , 0.739088 , 0.739085 ,
 0.739084 , 0.739084 , 0.739085 , 0.739085 , 0.739085 , 0.739085 , 0.739085 , 0.739085 , 0.739085 }
```

Out[299]=
```
{x → 0.739085 }
```

In[300]:=
```
bisection[Cos@y - y, {y, 0, 1.5}]
```

Out[300]=
```
{y → 0.739085 }
```

Rational approximation of $\sqrt{2}$:

In[301]:=
```
bisection[x * x - 2, {x, 1, 2}, AccuracyGoal → 20]
```

Out[301]=
$$\left\{x \to \frac{417\,402\,170\,410\,649\,030\,795}{295\,147\,905\,179\,352\,825\,856}\right\}$$

In[302]:=
```
(x /. %)^2 - 2.
```

Out[302]=
```
0.
```

1.6. Display

Gray level

Not all shades can be distinguished:

In[ ]:=
```
Row[GrayLevel /@ Range[0, 1, .05]]
```

Out[ ]=

Reduce interval so that all shades can be distinguished:

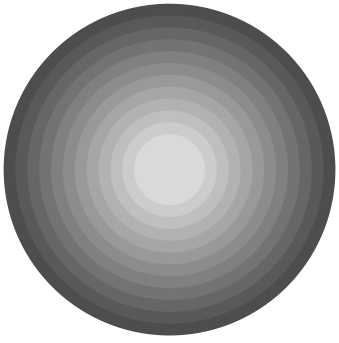
In[32]:=
```
grayLevelInterval = {.3, .85};
```

In[ ]:=
```
Row[GrayLevel /@ Range[Sequence @@ grayLevelInterval , .05]]
```

Out[ ]=

In[ ]:=
```
Style[Disk[{0, 0}, 1 - #], GrayLevel @ #] & /@ Range[Sequence @@ grayLevelInterval , .05] //
 Graphics[#, ImageSize → Small] &
```

Out[ ]=



Display positive numbers with fixed number of characters and without scientific notation

```
In[ ]:= 10. ^-5
        10. ^-6
        10.^5
        10.^6
```

```
Out[ ]= 0.00001
```

```
Out[ ]= 1. × 10^-6
```

```
Out[ ]= 100 000.
```

```
Out[ ]= 1. × 10^6
```

```
In[ ]:= PaddedForm[10. ^-6, 7]
        PaddedForm[10. ^-6, 7, ScientificNotationThreshold → {-1, 1} Infinity]
```

```
Out[ ]//PaddedForm=  1. × 10^-6
```

```
Out[ ]//PaddedForm=  0.000001
```

```
In[ ]:= StringPadLeft[#, 6] &@ ToString@PaddedForm[99 999.5 , 5]
        StringPadLeft[#, 6] &@ ToString@PaddedForm[99 999.499 , 5]
```

```
Out[ ]= 00000.
```

```
Out[ ]= 99999.
```

```
In[ ]:= StringPadLeft[#, 6] &@ ToString@PaddedForm[.00001 , 5]
        StringPadLeft[#, 6] &@ ToString@PaddedForm[.000009 , 5]
        StringPadLeft[#, 6] &@
         ToString@PaddedForm[.000009 , 5, ScientificNotationThreshold → {-1, 1} Infinity]
        StringPadLeft[#, 6] &@
         ToString@PaddedForm[Round[.0000050001 , 1. × 10^-5], 5,
           ScientificNotationThreshold → {-1, 1} Infinity]
```

```
Out[ ]= .00001
```

```
Out[ ]= . × 10
```

```
Out[ ]= 000009
```

```
Out[ ]= .00001
```

```
In[ ]:= StringPadLeft[#, 6] &@ ToString@PaddedForm[0.038018 , 5]
        StringPadLeft[#, 6] &@ ToString@PaddedForm[Round[0.038018 , 1. × 10^-5], 5]
```

```
Out[ ]= 038018
```

```
Out[ ]= .03802
```

```
In[ ]:= StringPadLeft [#, 6] &@
         ToString @ PaddedForm [Round [0.038 , 1. × 10 ^ - 5], 5,
           ScientificNotationThreshold   → {- 1, 1} Infinity ]
```

```
Out[ ]= 0.038
```

```
In[33]:= Clear @ constantLength ;
         constantLength [x_, length1_Integer , option : OptionsPattern []] :=
          StringPadLeft [#, length1 ] &@ Which [
            x ≥ 10 ^ (length1 - 1) - .5 ,
                ">",
            x > .5 × 10 ^ (- length1 + 1),
                StringReplace [#, " 0." → "  ."] &@
             ToString @ PaddedForm [Round [x, 1. × 10 ^ (- length1 + 1)], length1 - 1,
               ScientificNotationThreshold   → {- 1, 1} Infinity , option ],
            True ,
                "<"
           ]
```

```
In[ ]:= {constantLength [.123 , 6]}
        {constantLength [.1234 , 6]}
        {constantLength [.12345 , 6]}
        {constantLength [.123456 , 6]}
```

```
Out[ ]= {  .123}
```

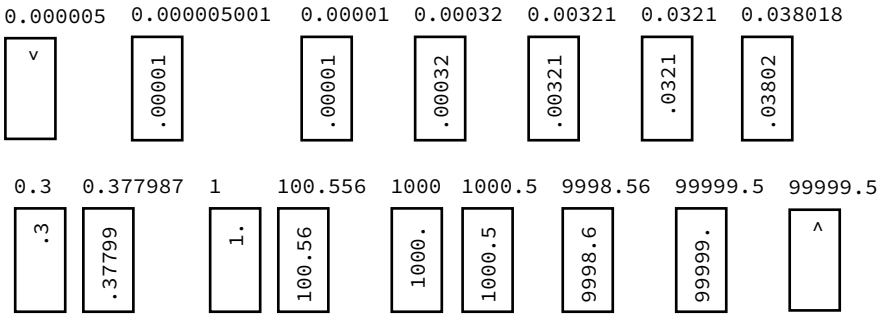
```
Out[ ]= { .1234}
```

```
Out[ ]= {.12345}
```

```
Out[ ]= {.12346}
```

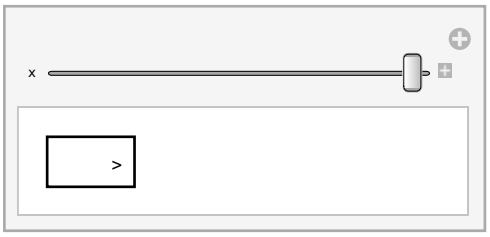
```
In[ ]:= Row [
         Column @ {DecimalForm [#],
             Rotate [#, Pi / 2] &@ Row [List @ constantLength [#, 6], Frame → True ]} & /@
          {.0000050 , .000005001 , .00001 , .00032 , .00321 , .0321 , 0.038018 , .3 , .377987 ,
           1, 100.5555 , 1000 , 1000.5 , 9998.555 , 99 999.499 , 99 999.5 }, " "]
```

```
Out[ ]=
```



In[ ]:=
```
With[{n = 5}, Manipulate [Row[List @ constantLength [10 ^ x, n], Frame → True], {x, -n, n}]]
```

Out[ ]=



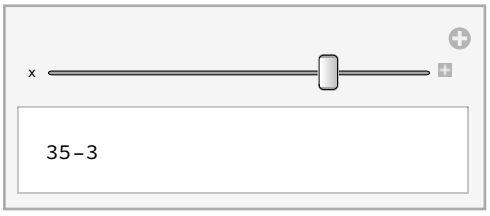
In[35]:=
```
constantLengthProbability  =
  With[{(*shift  number  point  to  the  right  to  eliminate  NumberPoint :*)sr1 = 2},
    Which[
     (*don't display  negative  probability *)
     Negative @ #,
          Style["<", FontColor → Red],
     (*too  small  number  => too  long  writing *)
     # < 10 ^(sr1 - 10),
          "<",
     True ,
          Composition [StringDrop [#, 3] &, StringJoin [#1 , (*"E",*)#2] & @@
             Apply[If[#1 == "  100", {"  10", ToString [ToExpression @ #2 + 1]}, {##}] &,
              ToString /@ {PaddedForm [# * 10 ^ sr1 , {2 + sr1 , 2 - sr1}, NumberPoint → ""],
                 PaddedForm [#2 - sr1 , 1, NumberSigns → {"-", "+"}]}] & @@
           MantissaExponent [#] &]@#]] &;
```

```
(*10E-2*)constantLengthProbability  [.1]
```

Out[ ]=
```
10-2
```

In[ ]:=
```
Manipulate [constantLengthProbability  [10 ^ x], {x, -9, 1, .01}]
```

Out[ ]=



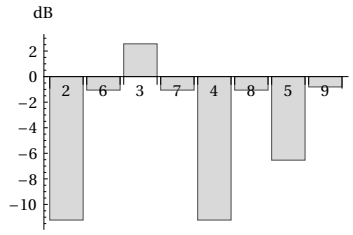
Decibel bar chart

Reference dB==0 is first element.

In[36]:=
```
Clear @ dBBarChart ;
dBBarChart [list1_ , option : OptionsPattern []] :=
 Composition [BarChart [First /@ #, option , AxesLabel → {None , "dB"},
      BarOrigin → Bottom , IntervalMarkers → None , ChartStyle → LightGray ,
      Axes → {True , True}, ChartLabels → Last /@ #, PlotRange → All] &,
   Flatten [#, 1] &@ Transpose @ Partition [Rest @ #, (Length @ # - 1) / 2] &,
   MapIndexed [{(*dB*)10 Log[10 , #1], only @ #2} &, # / First @ #] &]@ list1
```

In[ ]:=
```
dBBarChart [{578.51 , 43.664 , 1043.6647 , 43.664 , 128.42832 , 453.812 , 453.81215 ,
  453.81215 , 480.50 }]
```

Out[ ]=



2. Collision probability method for solving the integral stationary transport equation

The integral stationary transport equation is discretized in space, according to the collision probability approximation, see [1], ch. X, p. 181, with these assumptions:
- uniform properties of each region,
- isotropic collisions,
- monokinetic particles,
which yields a finite dimensional linear equation:

transport

In[37]:=
$$\text{transport} = V_i\,\Sigma_i\,\Phi_i == \sum_j P_{j\,i}\,V_j\,(\Sigma s_j\,\Phi_j + Q_j)$$

{1}

transport

Out[37]=
$$V_i\,\Sigma_i\,\Phi_i == \sum_j P_{j\,i}\,V_j\,(Q_j + \Sigma s_j\,\Phi_j)$$

with the all non negative

- V_i the volume of,
- Σ_i the inverse mean free path or total volumic (per unit volume) or ("macroscopic") cross section of,
- Φ_i the particle flux in,
- Σs_i (zero for a black region) the macroscopic scattering cross section of,
- Q_i the volumic spontaneous source in

region i,

- $P_{i\,j}$ the probability that a non leaking particle emitted in region i first collides (after rectilinear free flight) in region j.

Leaks can be represented implicitly by discounting the Σs_i and the Q_i or explicitly by black regions. The collision probability is normalized over the space partition, so that the matrix $P_{i\,j}$ (first index for row, second for column) is right stochastic [2]:

probabilityNorm

In[38]:=
```
probabilityNorm  = ∑_j P_ij == 1
```
{2}

probabilityNorm

Out[38]= $\sum_j P_{ij} == 1$

The P_{ij} depend only on the obstacles that free flying particles see: the Σ_i and the geometry, not the q_j, k_j.

In {1}, eliminate any non reactive ($\Sigma == 0$) region and for each remaining reactive ($\Sigma \neq 0$) region, introduce the collision rate $R == V\,\Sigma\,\Phi$, the multiplication factor $k == \Sigma s / \Sigma$ and the spontaneous source $q == V\,Q$:

markov

In[39]:=
```
markov =
 transport /. # → Expand[#] &[V_j (Σs_j Φ_j + Q_j)] /. {V_j Q_j → q_j, V_i Σ_i Φ_i → R_i, V_j Σs_j Φ_j → R_j k_j}
```
{3}

markov

Out[39]= $R_i == \sum_j P_{ji} (q_j + k_j R_j)$

{3} does not describe directly the evolution of particle numbers, it is just a detail of the reaction number in region i, according to all possible origins of incident particles.

{3} with $q_j == 0$ and $k_j == 0$ is also Markov eigenvalue equation [2], so I will keep this name. The equivalent form of left (row) eigenvector of the right stochastic matrix P_{ij} is:

In[ ]:=
```
markov /. HoldPattern[Indexed[P, ii_] * Plus[tt__]] :> HoldForm[Plus[tt] Indexed[P, ii]] //
  exchange[i, j]
```

Out[ ]= $R_j == \sum_i (q_i + k_i R_i) P_{ij}$

As a consequence of the monokinetic hypothesis, collision probabilities obey reciprocity, see [1], p. 182, 173:

reciprocity

In[40]:=
```
reciprocity  = HoldForm[V_i Σ_i P_ij == V_j Σ_j P_ji]
```
{4}

reciprocity

Out[40]= $V_i\,\Sigma_i\,P_{ij} == V_j\,\Sigma_j\,P_{ji}$

In particular, if two regions i, j have the same total cross section $V_i\,\Sigma_i == V_j\,\Sigma_j$, then $P_{ij} == P_{ji}$. This is true with a uniform mesh in a uniform medium.

Definition: a subset of regions $\{i_1 \ldots i_k\}$ is disconnected (from the others) if for all $j \notin \{i_1 \ldots i_k\}$, $P_{j\,i_k} == 0$. Then also by reciprocity {4} $P_{i_k\,j} == 0$. A single disconnected region is such that $P_{ii} == 1$. A subset of disconnected regions forms another problem of the same nature so that it can be eliminated without loss of generality.

3. Markov eigenvalue equation

3.1. Construction

In[ ]:=
```
expandSum[markov, 3]
```

Out[ ]= $R_i == P_{1i} (q_1 + k_1 R_1) + P_{2i} (q_2 + k_2 R_2) + P_{3i} (q_3 + k_3 R_3)$

```
Column @ Table[expandSum [markov , 3], {i , 3}]
```

```
R1 == P11 (q1 + k1 R1) + P21 (q2 + k2 R2) + P31 (q3 + k3 R3)
R2 == P12 (q1 + k1 R1) + P22 (q2 + k2 R2) + P32 (q3 + k3 R3)
R3 == P13 (q1 + k1 R1) + P23 (q2 + k2 R2) + P33 (q3 + k3 R3)
```

Straightforward manipulations

```
expandSum [markov , n]
```

$$R_i == \sum_j^n P_{j\,i}\,(q_j + k_j\,R_j)$$

```
MapAt [Expand , expandSum [markov , n], {2 , 1}]
```

$$R_i == \sum_j^n (P_{j\,i}\,q_j + k_j\,P_{j\,i}\,R_j)$$

```
Reverse [MapAt [Expand , expandSum [markov , n], {2 , 1}] /.
  HoldPattern [Sum[x_ + y_ , z_]] :> Sum[x , z] + Sum[y , z]]
```

$$\sum_j^n P_{j\,i}\,q_j + \sum_j^n k_j\,P_{j\,i}\,R_j == R_i$$

```
SubtractSides [#, #[[1 , 1]] + Last @ #] &[
 Reverse [MapAt [Expand , expandSum [markov , n], {2 , 1}] /.
   HoldPattern [Sum[x_ + y_ , z_]] :> Sum[x , z] + Sum[y , z]]]
```

$$-R_i + \sum_j^n k_j\,P_{j\,i}\,R_j == -\sum_j^n P_{j\,i}\,q_j$$

```
MapAt [
 # /. HoldPattern [Sum[x_ , {j , n}]] :> (x /. j → i) + Sum[Expand @ x , {j , 1 , i - 1}] +
      Sum[Expand @ x , {j , i + 1 , n}] &,
 SubtractSides [#, #[[1 , 1]] + Last @ #] &[
  Reverse [MapAt [Expand , expandSum [markov , n], {2 , 1}] /.
    HoldPattern [Sum[x_ + y_ , z_]] :> Sum[x , z] + Sum[y , z]]], {1 , 2}]
```

$$-R_i + k_i\,P_{i\,i}\,R_i + \sum_{j=1}^{-1+i} k_j\,P_{j\,i}\,R_j + \sum_{j=1+i}^{n} k_j\,P_{j\,i}\,R_j == -\sum_j^n P_{j\,i}\,q_j$$

In[41]:=
```
markov1 =
 Collect[
  MapAt[
   # /. HoldPattern[Sum[x_, {j, n}]] :> (x /. j → i) + Sum[Expand @ x, {j, 1, i - 1}] +
        Sum[Expand @ x, {j, i + 1, n}] &,
   SubtractSides[#, #[[1, 1]] + Last @ #] &[
    Reverse[MapAt[Expand, expandSum[markov, n], {2, 1}] /.
      HoldPattern[Sum[x_ + y_, z_]] :> Sum[x, z] + Sum[y, z]], {1, 2}], Ri]
```

Out[41]= $(-1 + k_i P_{ii}) R_i + \sum_{j=1}^{-1+i} k_j P_{ji} R_j + \sum_{j=1+i}^{n} k_j P_{ji} R_j == -\sum_{j}^{n} P_{ji} q_j$

yield a standard form for a linear system to solve:

In[ ]:=
```
Column @ Table[markov1 /. n → #, {i, #}] &@3
```

Out[ ]=
$$\begin{aligned}
&(-1 + k_1 P_{11}) R_1 + k_2 P_{21} R_2 + k_3 P_{31} R_3 == -P_{11} q_1 - P_{21} q_2 - P_{31} q_3\\
&k_1 P_{12} R_1 + (-1 + k_2 P_{22}) R_2 + k_3 P_{32} R_3 == -P_{12} q_1 - P_{22} q_2 - P_{32} q_3\\
&k_1 P_{13} R_1 + k_2 P_{23} R_2 + (-1 + k_3 P_{33}) R_3 == -P_{13} q_1 - P_{23} q_2 - P_{33} q_3
\end{aligned}$$

Extract the matrix and vector used for solving, depending on the solution method:

matrixForm

In[42]:=
```
Clear @ matrixVector ;
matrixVector[dim_, method → LinearSolve] :=
 Thread[{MapIndexed[Coefficient[#1, Apply[R#&, #2]] &, List @@ #], #2} & @@ # & /@ #] & @
  Table[markov1 /. n → dim, {i, dim}]

matrixVector[dim_, method → Solve] := {1, -1} matrixVector[dim, method → LinearSolve]

matrixVector[dim_, method → Inverse] := matrixVector[dim, method → LinearSolve]

matrixVector[dim_, method → FixedPoint] :=
 {#1 + DiagonalMatrix @ Table[1, dim], -#2} & @@ matrixVector[dim, method → LinearSolve]
```

{5}

3.2. Solution methods

In[ ]:=
```
MatrixForm /@ matrixVector[2, method → LinearSolve]
```

Out[ ]=
$$\left\{\begin{pmatrix} -1 + k_1 P_{11} & k_2 P_{21} \\ k_1 P_{12} & -1 + k_2 P_{22} \end{pmatrix}, \begin{pmatrix} -P_{11} q_1 - P_{21} q_2 \\ -P_{12} q_1 - P_{22} q_2 \end{pmatrix}\right\}$$

In[ ]:=
```
With[{dim = 2},
 Equal[#1 . #3, #2] & @@
  MatrixForm /@ Append[matrixVector[dim, method → LinearSolve], Table[Rj, {j, dim}]]]
```

Out[ ]=
$$\begin{pmatrix} -1 + k_1 P_{11} & k_2 P_{21} \\ k_1 P_{12} & -1 + k_2 P_{22} \end{pmatrix} . \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} == \begin{pmatrix} -P_{11} q_1 - P_{21} q_2 \\ -P_{12} q_1 - P_{22} q_2 \end{pmatrix}$$

```
In[ ]:= With[{dim = 2}, LinearSolve @@ matrixVector [dim, method → LinearSolve ]]
```

$$\text{Out[ ]}= \left\{\frac{-P_{11}\,q_1 - k_2\,P_{12}\,P_{21}\,q_1 + k_2\,P_{11}\,P_{22}\,q_1 - P_{21}\,q_2}{-1 + k_1\,P_{11} + k_1\,k_2\,P_{12}\,P_{21} + k_2\,P_{22} - k_1\,k_2\,P_{11}\,P_{22}},\ \frac{P_{12}\,q_1 + k_1\,P_{12}\,P_{21}\,q_2 + P_{22}\,q_2 - k_1\,P_{11}\,P_{22}\,q_2}{1 - k_1\,P_{11} - k_1\,k_2\,P_{12}\,P_{21} - k_2\,P_{22} + k_1\,k_2\,P_{11}\,P_{22}}\right\}$$

```
In[ ]:= % /. {k1 → 0, k2 → 0}
```

$$\text{Out[ ]}= \{P_{11}\,q_1 + P_{21}\,q_2,\ P_{12}\,q_1 + P_{22}\,q_2\}$$

```
In[ ]:= matrixVector [2, method → Solve]
```

$$\text{Out[ ]}= \{\{\{-1 + k_1\,P_{11},\ k_2\,P_{21}\},\ \{k_1\,P_{12},\ -1 + k_2\,P_{22}\}\},\ \{P_{11}\,q_1 + P_{21}\,q_2,\ P_{12}\,q_1 + P_{22}\,q_2\}\}$$

```
In[47]:= Clear @ markov2 ;
         markov2 @ dim_ := Identity [#1 . #3 + #2 == #4] & @@
           MatrixForm /@ Join[matrixVector [dim, method → Solve],
             {Table[ Ri, {i, dim}], Table[ 0, dim]}]
```

```
In[ ]:= markov2 @ 2
```

$$\text{Out[ ]}= \begin{pmatrix} -1 + k_1\,P_{11} & k_2\,P_{21} \\ k_1\,P_{12} & -1 + k_2\,P_{22} \end{pmatrix} \cdot \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} + \begin{pmatrix} P_{11}\,q_1 + P_{21}\,q_2 \\ P_{12}\,q_1 + P_{22}\,q_2 \end{pmatrix} == \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

```
In[ ]:= With[{dim = 2},
          only @ Solve [#1 . #3 + #2 == #4 , #3] & @@
           Join[matrixVector [dim, method → Solve], {Table[ Ri, {i, dim}], Table[ 0, dim]}]]
```

$$\text{Out[ ]}= \left\{R_1 \to -\frac{P_{11}\,q_1 + k_2\,P_{12}\,P_{21}\,q_1 - k_2\,P_{11}\,P_{22}\,q_1 + P_{21}\,q_2}{-1 + k_1\,P_{11} + k_1\,k_2\,P_{12}\,P_{21} + k_2\,P_{22} - k_1\,k_2\,P_{11}\,P_{22}},\right.$$
$$\left. R_2 \to -\frac{P_{12}\,q_1 + k_1\,P_{12}\,P_{21}\,q_2 + P_{22}\,q_2 - k_1\,P_{11}\,P_{22}\,q_2}{-1 + k_1\,P_{11} + k_1\,k_2\,P_{12}\,P_{21} + k_2\,P_{22} - k_1\,k_2\,P_{11}\,P_{22}}\right\}$$

```
In[ ]:= % /. {k1 → 0, k2 → 0}
```

$$\text{Out[ ]}= \{R_1 \to P_{11}\,q_1 + P_{21}\,q_2,\ R_2 \to P_{12}\,q_1 + P_{22}\,q_2\}$$

```
In[ ]:= MatrixForm /@ matrixVector [2, method → Inverse ]
```

$$\text{Out[ ]}= \left\{\begin{pmatrix} -1 + k_1\,P_{11} & k_2\,P_{21} \\ k_1\,P_{12} & -1 + k_2\,P_{22} \end{pmatrix}, \begin{pmatrix} -P_{11}\,q_1 - P_{21}\,q_2 \\ -P_{12}\,q_1 - P_{22}\,q_2 \end{pmatrix}\right\}$$

```
In[ ]:= Inverse @ First @ #. Last @ # & @@ matrixVector [2, method → Inverse ] // Together
```

$$\text{Out[ ]}= \left\{\frac{-P_{11}\,q_1 - k_2\,P_{12}\,P_{21}\,q_1 + k_2\,P_{11}\,P_{22}\,q_1 - P_{21}\,q_2}{-1 + k_1\,P_{11} + k_1\,k_2\,P_{12}\,P_{21} + k_2\,P_{22} - k_1\,k_2\,P_{11}\,P_{22}},\ \frac{-P_{12}\,q_1 - k_1\,P_{12}\,P_{21}\,q_2 - P_{22}\,q_2 + k_1\,P_{11}\,P_{22}\,q_2}{-1 + k_1\,P_{11} + k_1\,k_2\,P_{12}\,P_{21} + k_2\,P_{22} - k_1\,k_2\,P_{11}\,P_{22}}\right\}$$

Examine the inverse matrix:

```
In[ ]:= Inverse @ First @ matrixVector [2, method → Inverse ] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{-1+k_2 P_{22}}{1-k_1 P_{11}-k_1 k_2 P_{12} P_{21}-k_2 P_{22}+k_1 k_2 P_{11} P_{22}} & -\frac{k_2 P_{21}}{1-k_1 P_{11}-k_1 k_2 P_{12} P_{21}-k_2 P_{22}+k_1 k_2 P_{11} P_{22}} \\ -\frac{k_1 P_{12}}{1-k_1 P_{11}-k_1 k_2 P_{12} P_{21}-k_2 P_{22}+k_1 k_2 P_{11} P_{22}} & \frac{-1+k_1 P_{11}}{1-k_1 P_{11}-k_1 k_2 P_{12} P_{21}-k_2 P_{22}+k_1 k_2 P_{11} P_{22}} \end{pmatrix}$$

```
In[ ]:= (Inverse @ First @ matrixVector [3, method → Inverse ])[[1, 1]] // Simplify
```

Out[ ]=

$$(1 - k_3 P_{33} + k_2 (-k_3 P_{23} P_{32} + P_{22} (-1 + k_3 P_{33}))) /$$
$$(k_1 k_3 P_{13} ((1 - k_2 P_{22}) P_{31} + k_2 P_{21} P_{32}) + k_2 k_3 P_{23} (P_{32} + k_1 (P_{12} P_{31} - P_{11} P_{32})) +$$
$$(1 - k_2 P_{22} + k_1 (-k_2 P_{12} P_{21} + P_{11} (-1 + k_2 P_{22}))) (-1 + k_3 P_{33}))$$

```
In[ ]:= With[{dim = 3}, monomialCollect [Table[k_i, {i, dim}]]@
          (Inverse @ First @ matrixVector [dim, method → Inverse ])[[1, 1]]
```

Out[ ]=

$$(1 - k_2 P_{22} - k_3 P_{33} + k_2 k_3 (-P_{23} P_{32} + P_{22} P_{33})) /$$
$$(-1 + k_1 P_{11} + k_2 P_{22} + k_1 k_2 (P_{12} P_{21} - P_{11} P_{22}) + k_3 P_{33} + k_1 k_3 (P_{13} P_{31} - P_{11} P_{33}) + k_2 k_3 (P_{23} P_{32} - P_{22} P_{33}) +$$
$$k_1 k_2 k_3 (-P_{13} P_{22} P_{31} + P_{12} P_{23} P_{31} + P_{13} P_{21} P_{32} - P_{11} P_{23} P_{32} - P_{12} P_{21} P_{33} + P_{11} P_{22} P_{33}))$$

```
In[ ]:= Inverse @ First @ matrixVector [3, method → Inverse ] /. {k_2 → 0} // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{1-k_3 P_{33}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} & 0 & \frac{k_3 P_{31}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} \\ \frac{k_1 P_{12}+k_1 k_3 P_{13} P_{32}-k_1 k_3 P_{12} P_{33}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} & \frac{1-k_1 P_{11}-k_1 k_3 P_{13} P_{31}-k_3 P_{33}+k_1 k_3 P_{11} P_{33}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} & \frac{k_1 k_3 P_{12} P_{31}+k_3 P_{32}-k_1 k_3 P_{11} P_{32}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} \\ \frac{k_1 P_{13}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} & 0 & \frac{1-k_1 P_{11}}{k_1 k_3 P_{13} P_{31}+(1-k_1 P_{11})(-1+k_3 P_{33})} \end{pmatrix}$$

Black regions are equivalent to leak. This is a very simple case:

```
In[ ]:= Inverse @ First @ matrixVector [3, method → Inverse ] /. {k_2 → 0, k_3 → 0} // Simplify //
          MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{1}{-1+k_1 P_{11}} & 0 & 0 \\ \frac{k_1 P_{12}}{-1+k_1 P_{11}} & -1 & 0 \\ \frac{k_1 P_{13}}{-1+k_1 P_{11}} & 0 & -1 \end{pmatrix}$$

```
In[ ]:= matrixVector [2, method → FixedPoint ]
```

Out[ ]= {{{$k_1 P_{11}$, $k_2 P_{21}$}, {$k_1 P_{12}$, $k_2 P_{22}$}}, {$P_{11} q_1 + P_{21} q_2$, $P_{12} q_1 + P_{22} q_2$}}

FixedPoint

```
In[ ]:= With[{dim = 2},
          With[{solutionVector  = Table[R_j, {j, dim}]},
           MatrixForm @ solutionVector  == Plus[Dot[MatrixForm @ #1, MatrixForm @ solutionVector ],
                MatrixForm @ #2] & @@ matrixVector [dim, method → FixedPoint ]]]
```

{6}

Out[ ]=

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} == \begin{pmatrix} k_1 P_{11} & k_2 P_{21} \\ k_1 P_{12} & k_2 P_{22} \end{pmatrix} . \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} + \begin{pmatrix} P_{11} q_1 + P_{21} q_2 \\ P_{12} q_1 + P_{22} q_2 \end{pmatrix}$$

Iterative solution:

```
In[ ]:= With[{dim = 2},
          With[{matrixVector1 = matrixVector[dim, method → FixedPoint]},
           With[
            {function1 = Function[Plus[Dot[matrix1, #], vector1]] /.
                {matrix1 → First@matrixVector1, vector1 → Last@matrixVector1}},
            NestList[function1, Table[0, dim], 3]
           ]]] // Column
```

```
Out[ ]= {0, 0}

{P11 q1 + P21 q2, P12 q1 + P22 q2}

{P11 q1 + P21 q2 + k1 P11 (P11 q1 + P21 q2) + k2 P21 (P12 q1 + P22 q2),
  P12 q1 + P22 q2 + k1 P12 (P11 q1 + P21 q2) + k2 P22 (P12 q1 + P22 q2)}

{P11 q1 + P21 q2 + k1 P11 (P11 q1 + P21 q2 + k1 P11 (P11 q1 + P21 q2) + k2 P21 (P12 q1 + P22 q2)) +
   k2 P21 (P12 q1 + P22 q2 + k1 P12 (P11 q1 + P21 q2) + k2 P22 (P12 q1 + P22 q2)),
  P12 q1 + P22 q2 + k1 P12 (P11 q1 + P21 q2 + k1 P11 (P11 q1 + P21 q2) + k2 P21 (P12 q1 + P22 q2)) +
   k2 P22 (P12 q1 + P22 q2 + k1 P12 (P11 q1 + P21 q2) + k2 P22 (P12 q1 + P22 q2))}
```

complexity

3.3. Complexity

Only LinearSolve *on numbers* is expected to work for large dimension. The general solution by the inverse matrix is only practical for small dimension (depending on available storage):

```
In[ ]:= Clear@bc;
        bc@n_Integer := (bc@n = ByteCount[LinearSolve @@ matrixVector @n])
```

```
In[ ]:= bc@2
```

```
... LinearSolve : Argument 2 at position 1 is not a non-empty rectangular matrix.
```

```
Out[ ]= 64
```

```
In[ ]:= bc@3
```

```
... LinearSolve : Argument 3 at position 1 is not a non-empty rectangular matrix.
```

```
Out[ ]= 64
```

```
In[ ]:= bc@4
```

```
... LinearSolve : Argument 4 at position 1 is not a non-empty rectangular matrix.
```

```
Out[ ]= 64
```

```
In[ ]:= bc@5
```

```
... LinearSolve : Argument 5 at position 1 is not a non-empty rectangular matrix.
```

```
Out[ ]= 64
```

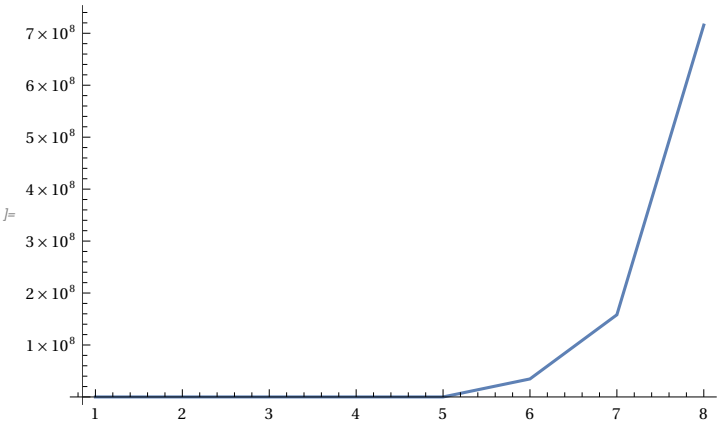
```
In[ ]:= Timing@bc@6
```

```
... LinearSolve : Argument 6 at position 1 is not a non-empty rectangular matrix.
```

```
Out[ ]= {0.002879, 64}
```

```
In[ ]:= (*Timing @bc@7*)
        bc@7 = 34 651 728 ;

In[ ]:= bc@8 = 158 136 744 ;

In[ ]:= bc@9 = 715 689 016 ;

In[ ]:= ListPlot[Table[Evaluate @bc@i, {i, 2, 9}], PlotRange → All, Joined → True]
```

Out[ ]=



A uniform stationary distribution exists for any dimension when all collision probabilities are equal (but the k_i, q_i are not necessarily equal):

```
In[ ]:= With[{dim = 3}, markov2 @dim /. P__ → 1 / dim]
```

Out[ ]=

$$\begin{pmatrix} -1+\frac{k_1}{3} & \frac{k_2}{3} & \frac{k_3}{3} \\ \frac{k_1}{3} & -1+\frac{k_2}{3} & \frac{k_3}{3} \\ \frac{k_1}{3} & \frac{k_2}{3} & -1+\frac{k_3}{3} \end{pmatrix} \cdot \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} + \begin{pmatrix} \frac{q_1}{3}+\frac{q_2}{3}+\frac{q_3}{3} \\ \frac{q_1}{3}+\frac{q_2}{3}+\frac{q_3}{3} \\ \frac{q_1}{3}+\frac{q_2}{3}+\frac{q_3}{3} \end{pmatrix} == \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

```
In[ ]:= (# /. x_ / y_ :→ -x / -y) & /@
          Simplify @With[{dim = 3}, LinearSolve @@ matrixVector[dim, method → LinearSolve] /.
            P__ → 1 / dim]
```

Out[ ]=

$$\left\{\frac{q_1+q_2+q_3}{3-k_1-k_2-k_3}, \frac{q_1+q_2+q_3}{3-k_1-k_2-k_3}, \frac{q_1+q_2+q_3}{3-k_1-k_2-k_3}\right\}$$

```
In[ ]:= With[{dim = 3},
          Map[Total, Inverse[First @matrixVector[dim, method → Inverse] /. P__ → 1 / dim]]] //
         Simplify
```

Out[ ]=

$$\left\{\frac{3}{-3+k_1+k_2+k_3}, \frac{3}{-3+k_1+k_2+k_3}, \frac{3}{-3+k_1+k_2+k_3}\right\}$$

```
In[ ]:= With[{dim = 7}, Inverse[First @matrixVector[dim, method → Inverse] /. P__ → 1 / dim][[1, 1]]] //
         Simplify
```

Out[ ]=

$$-\frac{-7+k_2+k_3+k_4+k_5+k_6+k_7}{-7+k_1+k_2+k_3+k_4+k_5+k_6+k_7}$$

criticality

3.4. Non negativity and criticality

For the Markov eigenvalue equation {5} to represent the integral transport equation {1}, its solution must be non negative.

Global properties:

globalProperties

In[48]:=
```
Clear @ BracketingBar ;
|R_| := Σ_i R_i
Clear @ OverBar ;
(*parenthesis  required  for  AutoGeneratedPackage  *)k̄_ := (Σ_i k_i R_i) / |R|
```
{7}

By non negativity, |R| == 0 if and only if for all i, R_i == 0 (the function R → |R| is a norm).

The global (R-weighted average) multiplication factor $\bar{k}$ is not a parameter as it depends on the unknown R_i. Yet by non negativity $0 \le \text{Min}[k_i] \le \bar{k} \le \text{Max}[k_i]$ where the k_i are parameters.

Sum {3} over i, use {2}:

SumRi

In[ ]:=
```
Map[Σ_i # &, markov] /.
  HoldPattern[Σ_i (Σ_j x_)] :> Σ_j (Σ_i x) /.
 Σ_i P_ji x_ /; FreeQ[x, i] -> (Σ_i P_ji) x /. Apply[Rule, exchange[i, j] @ probabilityNorm] /.
HoldPattern[Σ_j (x_ + y_)] :> exchange[i, j][Σ_j x + Σ_j y]
```
{8}

SumRi

Out[ ]=
$$\sum_i R_i == \sum_i q_i + \sum_i k_i R_i$$

Rewrite {8} with {7}:

globalBalance

In[ ]:=
```
(% /. {Σ_i k_i R_i :> HoldForm @ k̄ × HoldForm @ |R|, Σ_j_ R_j_ :> HoldForm @ |R|} //
  SubtractSides[#, k̄ HoldForm @ |R|] &) /. x_ - x_ y_ -> x (1 - HoldForm @ y) //
 MultiplySides[#, 1 / (1 - k̄), Assumptions -> k̄ < 1] &
```
{9}

globalBalance

Out[ ]=
$$|R| == \frac{|q|}{1 - \bar{k}}$$

$\bar{k}$ == 1 is the critical condition. It is logically equivalent to |q| == 0 and |R| can take any value. Quasi-criticality is when $\bar{k}$ is close to but lower than one. |R| / |q| is the spontaneous source amplification factor.

{9}, equivalent to {5} in dimension 1, solves the single region Markov eigenvalue equation:

```
In[ ]:= AddSides[MapAt[# /. x_. y_ :> HoldForm[y x] &,
          MultiplySides[MapAt[# /. x_ / y_ :> -x / -y &, %, 2], -1 + k̄, Assumptions → k̄ < 1], 1], |q|] /.
        x_ + y_ :> HoldForm[y + x]
```

$$\text{Out[ ]=}\quad \left(-1+\bar{k}\right)|R|+|q|==0$$

The diagonal coefficients of the stochastic matrix must be non positive:

```
In[ ]:= markov2 @ 3
```

$$\text{Out[ ]=}\quad \begin{pmatrix} -1+k_1 P_{11} & k_2 P_{21} & k_3 P_{31} \\ k_1 P_{12} & -1+k_2 P_{22} & k_3 P_{32} \\ k_1 P_{13} & k_2 P_{23} & -1+k_3 P_{33} \end{pmatrix}.\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}+\begin{pmatrix} P_{11} q_1+P_{21} q_2+P_{31} q_3 \\ P_{12} q_1+P_{22} q_2+P_{32} q_3 \\ P_{13} q_1+P_{23} q_2+P_{33} q_3 \end{pmatrix}==\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

If $P_{11}==1$ then region 1 is disconnected and eliminated. Thus $P_{11}<1$. If $k_1>1/P_{11}>1$, then a non negative solution does not exist and the region 1 alone is super-critical but $k_1\geq 1$ is possible if the other $k_{i\neq 1}$ are so small that overall $\bar{k}<1$.

4. Collision probability model

eliminate

4.1. Eliminate diagonal collision probabilities

Knowing the non diagonal collision probabilities P_{ij}, $i\neq j$, the diagonal ones P_{ii} can be determined by the normalization {2}:

```
With[{dim = 3},
 Print @ Column @ Table[expandSum[probabilityNorm , dim], {i, dim}];
 First @ Solve[Table[ReleaseHold @ expandSum[probabilityNorm , dim], {i, dim}],
   Table[Pii, {i, dim}]]]
```

$$P_{11}+P_{12}+P_{13}==1$$
$$P_{21}+P_{22}+P_{23}==1$$
$$P_{31}+P_{32}+P_{33}==1$$

$$\text{Out[ ]=}\quad \{P_{11}\to 1-P_{12}-P_{13},\ P_{22}\to 1-P_{21}-P_{23},\ P_{33}\to 1-P_{31}-P_{32}\}$$

Equivalently:

```
In[ ]:= With[{dim = 3}, Map[Total @ # /. x_ + Pi_i_ :> Pii → 1 - x &, Table[Pij, {i, dim}, {j, dim}]]]
```

$$\text{Out[ ]=}\quad \{P_{11}\to 1-P_{12}-P_{13},\ P_{22}\to 1-P_{21}-P_{23},\ P_{33}\to 1-P_{31}-P_{32}\}$$

or:

```
In[ ]:= With[{dim = 3}, Table[Pii → 1 - Sum[Pij, {j, 1, i - 1}] - Sum[Pij, {j, i + 1, dim}], {i, dim}]]
```

$$\text{Out[ ]=}\quad \{P_{11}\to 1-P_{12}-P_{13},\ P_{22}\to 1-P_{21}-P_{23},\ P_{33}\to 1-P_{31}-P_{32}\}$$

Warn if normalization is impossible:

In[52]:=
```
solveDiagonal::negativeProbability =
  "probabilities can't be normalized at line `1`";
Clear@solveDiagonal;
solveDiagonal[dim_Integer, OptionsPattern[checkNonNegative → True]] :=
 Table[P_ii → 1 - With[{position = i}, If[Not@OptionValue@checkNonNegative, #,
         If[# ≤ 1, #, Message[solveDiagonal::negativeProbability, position];
          1]]] &[Sum_{j=1}^{i-1} P_ij + Sum_{j=i+1}^{dim} P_ij], {i, dim}]
```

In[ ]:=
```
Column@solveDiagonal[3, checkNonNegative → False]
```

Out[ ]=
$$P_{11} \to 1 - P_{12} - P_{13}$$
$$P_{22} \to 1 - P_{21} - P_{23}$$
$$P_{33} \to 1 - P_{31} - P_{32}$$

In[ ]:=
```
Column@solveDiagonal[3, checkNonNegative → True]
```

Out[ ]=
$$P_{11} \to 1 - \text{If}[P_{12} + P_{13} \le 1, P_{12} + P_{13}, \text{Message}[\text{solveDiagonal::negativeProbability}, 1]; 1]$$
$$P_{22} \to 1 - \text{If}[P_{21} + P_{23} \le 1, P_{21} + P_{23}, \text{Message}[\text{solveDiagonal::negativeProbability}, 2]; 1]$$
$$P_{33} \to 1 - \text{If}[P_{31} + P_{32} \le 1, P_{31} + P_{32}, \text{Message}[\text{solveDiagonal::negativeProbability}, 3]; 1]$$

In[ ]:=
```
% /. {P21 → .6, P23 → .7}
```

```
••• solveDiagonal : probabilities can't be normalized at line 2
```

Out[ ]=
$$P_{11} \to 1 - \text{If}[P_{12} + P_{13} \le 1, P_{12} + P_{13}, \text{Message}[\text{solveDiagonal::negativeProbability}, 1]; 1]$$
$$P_{22} \to 0$$
$$P_{33} \to 1 - \text{If}[P_{31} + P_{32} \le 1, P_{31} + P_{32}, \text{Message}[\text{solveDiagonal::negativeProbability}, 3]; 1]$$

4.2. Point regions

In some large uniform medium, the pile, indexed by 1, consider two infinitely small regions indexed by `i, j ≥ 2, i ≠ j` and let $d_{ij} > 0$ be the distance between them.

By isotropy and definition of cross section, and considering that particle free flight is rectilinear (before first collision), P_{ij} must be proportional to $V_j \Sigma_j / (4 \pi d_{ij}^2)$, the fraction of the sphere intercepted by the target, with a dimensionless proportionality coefficient given by the Beer-Lambert- Bouguer attenuation law [3] (one if the medium is non reactive):

Pij

In[55]:=
```
probabilityModel0 = P_ij → V_j Σ_j / (4 π d_ij^2) expM[Σ_1 d_ij]
```
{10}

Pij

Out[55]=
$$P_{ij} \to \frac{\text{expM}[d_{ij}\,\Sigma_1]\,V_j\,\Sigma_j}{4\,\pi\,d_{ij}^2}$$

where `expM` has been defined in section 1 and its argument is named the *optical thickness*.

More generally, a non uniform medium is decomposed step by step in `n` smaller uniform media pathwise (along the path). Let `x` be the rectilinear ordinate from region `i` to region `j`, with origin at `i`. Let `x[k]` be the ordinate of the pathwise `k`-th interface and, if `k ≠ 0`, `m[k]` be the medium just before `x[k]` (in particular, the segment `[0, x[1]]` spans region `m[1]`). Decompose optical

thickness and introduce optical path:

```
In[56]:= boundaryCondition = {x[0] → 0, x[n] → d_ij}

         opticalThickness = HoldForm[Sum[Σ_m[k] (x[k] - x[-1 + k]), {k, 1, n}] /. #] & @@ boundaryCondition

         opticalPath = opticalThickness /. Sum → Table

         path = opticalPath /. Σ_ d_ :> d
```

Out[56]= $\{x[0] \to 0,\ x[n] \to d_{ij}\}$

Out[57]= $\sum_{k=1}^{n} \Sigma_{m[k]} (x[k] - x[-1 + k]) /. \{x[0] \to 0,\ x[n] \to d_{ij}\}$

Out[58]= $\text{Table}\left[\Sigma_{m[k]} (x[k] - x[-1 + k]), \{k, 1, n\}\right] /. \{x[0] \to 0,\ x[n] \to d_{ij}\}$

Out[59]= $\text{Table}\left[x[k] - x[-1 + k], \{k, 1, n\}\right] /. \{x[0] \to 0,\ x[n] \to d_{ij}\}$

In particular:

```
In[ ]:= ReleaseHold[opticalThickness /. n → #] & /@ Range @ 3 // Column
```

Out[ ]=
$d_{ij}\ \Sigma_{m[1]}$
$\Sigma_{m[2]} (d_{ij} - x[1]) + \Sigma_{m[1]}\ x[1]$
$\Sigma_{m[1]}\ x[1] + \Sigma_{m[3]} (d_{ij} - x[2]) + \Sigma_{m[2]} (-x[1] + x[2])$

```
In[ ]:= ReleaseHold[opticalThickness /. Σ_ → 1] == d_ij
```

Out[ ]= True

```
In[ ]:= ReleaseHold[opticalPath /. n → #] & /@ Range @ 3 // Column
```

Out[ ]=
$\{d_{ij}\ \Sigma_{m[1]}\}$
$\{\Sigma_{m[1]}\ x[1],\ \Sigma_{m[2]} (d_{ij} - x[1])\}$
$\{\Sigma_{m[1]}\ x[1],\ \Sigma_{m[2]} (-x[1] + x[2]),\ \Sigma_{m[3]} (d_{ij} - x[2])\}$

```
In[ ]:= ReleaseHold[path /. n → 3]
```

Out[ ]= $\{x[1],\ -x[1] + x[2],\ d_{ij} - x[2]\}$

For $n \geq 2$, pull apart the contributions of regions i, j:

opticalThickness1

```
In[60]:= opticalThickness1 = ReleaseHold[opticalThickness /. HoldPattern @ Sum[x_, {k, 1, n}] :>
           Plus @@ {x /. k → 1, Identity[Sum[x, {k, 2, n - 1}]], x /. k → n}] /.
         HoldPattern @ HoldForm[Table[x_, y_]] :> Table[x, y]
```

{11}

opticalThickness1

Out[60]= $\sum_{k=2}^{-1+n} \Sigma_{m[k]} (-x[-1 + k] + x[k]) + \Sigma_{m[1]}\ x[1] + \Sigma_{m[n]} (d_{ij} - x[-1 + n])$

In[61]:=
```
opticalPath1 = ReleaseHold[opticalPath /. HoldPattern@Table[x_, {k, 1, n}] :>
    {x /. k → 1, HoldForm[Sequence @@ Table[x, {k, 2, n - 1}]], x /. k → n}] /.
  HoldPattern@HoldForm[Table[x_, y_]] :> Table[x, y]
```
{12}

Out[61]= $\{\Sigma_{m[1]}\, x[1],\ \text{Sequence @@ Table}\left[\Sigma_{m[k]}\,(x[k]-x[-1+k]),\ \{k,\ 2,\ n-1\}\right],\ \Sigma_{m[n]}\,(d_{ij}-x[-1+n])\}$

In particular:

In[ ]:=
```
opticalThickness1 /. n → 2
ReleaseHold[opticalPath1 /. n → 2]
```

Out[ ]= $\Sigma_{m[2]}\,(d_{ij}-x[1])+\Sigma_{m[1]}\,x[1]$

Out[ ]= $\{\Sigma_{m[1]}\,x[1],\ \Sigma_{m[2]}\,(d_{ij}-x[1])\}$

In[ ]:=
```
opticalThickness1 /. n → 3
ReleaseHold[opticalPath1 /. n → 3]
```

Out[ ]= $\Sigma_{m[1]}\,x[1]+\Sigma_{m[3]}\,(d_{ij}-x[2])+\Sigma_{m[2]}\,(-x[1]+x[2])$

Out[ ]= $\{\Sigma_{m[1]}\,x[1],\ \Sigma_{m[2]}\,(-x[1]+x[2]),\ \Sigma_{m[3]}\,(d_{ij}-x[2])\}$

Use {11} in {10}:

Pij1

In[62]:=
```
probabilityModel1 = probabilityModel0 /. Σ1 d_ij → opticalThickness1
```
{13}

Pij1

Out[62]= $P_{ij} \to \frac{1}{4\,\pi\,d_{ij}^{\,2}}\,\text{expM}\left[\sum_{k=2}^{-1+n}\Sigma_{m[k]}\,(-x[-1+k]+x[k])+\Sigma_{m[1]}\,x[1]+\Sigma_{m[n]}\,(d_{ij}-x[-1+n])\right]V_j\,\Sigma_j$

In particular:

In[ ]:=
```
probabilityModel1 /. n → 2
```

Out[ ]= $P_{ij} \to \frac{\text{expM}\left[\Sigma_{m[2]}\,(d_{ij}-x[1])+\Sigma_{m[1]}\,x[1]\right]V_j\,\Sigma_j}{4\,\pi\,d_{ij}^{\,2}}$

{13} is just eq. 17, p. 181 of [1] applied to point regions or the Beer-Lambert- Bouguer law for inhomogeneous medium.

4.3. Finite regions

Each path extremity can be anywhere inside each finite region, d_{ij} is not exactly defined. Nevertheless, the collision probability is still estimated with {13}, where each extremity at some arbitrary *collapse* point in the region, which is imbedded in the medium. We will count as if all the matter of the region had collapsed in one point. In some cases, the best estimate is obtained by symmetry. This simplistic approximation may suffice to obtain tendencies with respect to some perturbation (change). The usual nearly exact method is based on volume integrals, see [1], ch. X, p. 171, 181.

Pij2

In[63]:=
```
boundaryCondition1  = {m[1] → i, m[n] → j}

opticalThickness2  = opticalThickness1  /. boundaryCondition1

opticalPath2  = opticalPath1  /. boundaryCondition1

probabilityModel2  = probabilityModel1  /. boundaryCondition1
```
{14}

Pij2

Out[63]= $\{m[1] \to i,\ m[n] \to j\}$

Pij2

Out[64]= $\sum_{k=2}^{-1+n} \Sigma_{m[k]} \left(-x[-1+k] + x[k]\right) + \Sigma_i\, x[1] + \Sigma_j \left(d_{ij} - x[-1+n]\right)$

Pij2

Out[65]= $\left\{\Sigma_i\, x[1],\ \text{Sequence @@ Table}\left[\Sigma_{m[k]} \left(x[k] - x[-1+k]\right),\ \{k,\ 2,\ n-1\}\right],\ \Sigma_j \left(d_{ij} - x[-1+n]\right)\right\}$

Pij2

Out[66]= $P_{ij} \to \dfrac{\text{expM}\left[\sum_{k=2}^{-1+n} \Sigma_{m[k]} \left(-x[-1+k] + x[k]\right) + \Sigma_i\, x[1] + \Sigma_j \left(d_{ij} - x[-1+n]\right)\right] V_j\, \Sigma_j}{4\,\pi\, {d_{ij}}^2}$

In particular:

In[ ]:=
```
probabilityModel2  /. n → 3
```

Out[ ]= $P_{ij} \to \dfrac{\text{expM}\left[\Sigma_i\, x[1] + \Sigma_j \left(d_{ij} - x[2]\right) + \Sigma_{m[2]} \left(-x[1] + x[2]\right)\right] V_j\, \Sigma_j}{4\,\pi\, {d_{ij}}^2}$

Apply {14} reciprocally from j to i, reversing the orientation of x, m:

In[ ]:=
```
exchange[i, j]@probabilityModel2  /. {
   HoldPattern[x[k_]] → exchange[i, j]@d_ij - x[n - k],
   (*x[1]→ exchange[i,j]@d_ij-x[n-1],
   x[n-1]→ exchange[i,j]@d_ij-x[1],*)
   m[k_] :→ m[n - k + 1]}
```

Out[ ]= $P_{ji} \to \dfrac{1}{4\,\pi\, {d_{ji}}^2}\, \text{expM}\left[\sum_{k=2}^{-1+n} \Sigma_{m[1-k+n]} \left(-x[-k+n] + x[1-k+n]\right) + \Sigma_i\, x[1] + \Sigma_j \left(d_{ji} - x[-1+n]\right)\right] V_i\, \Sigma_i$

Reindex and use that this does not change the summation bounds:

In[ ]:=
```
probabilityModel2r  =
  % /. HoldPattern[Sum[expr_, {k, bounds__}]] :→ Sum[expr /. k → n - k + 1, {k, bounds}]
```

Out[ ]= $P_{ji} \to \dfrac{\text{expM}\left[\sum_{k=2}^{-1+n} \Sigma_{m[k]} \left(-x[-1+k] + x[k]\right) + \Sigma_i\, x[1] + \Sigma_j \left(d_{ji} - x[-1+n]\right)\right] V_i\, \Sigma_i}{4\,\pi\, {d_{ji}}^2}$

Check reciprocity {4}:

```
In[ ]:= Simplify[ReleaseHold[reciprocity /. probabilityModel2 /. probabilityModel2r],
          d_ij == d_ji]
Out[ ]= True
```

4.4. Thick target region

Definition: target region i is thick if its optical thickness is greater than one: $\Sigma_i \, x[1] > 1$.

Rearrange {14} with xExpM :

Pij3

```
In[67]:= probabilityModel3 =
           probabilityModel2 /. HoldPattern @ expM[x_Sum + Σ_i y_ + Σ_j z_] → expM[x + Σ_i y] xExpM[Σ_j z] / Σ_j z
```

{15}

Pij3

$$\text{Out[67]=}\quad P_{ij} \to \frac{\text{expM}\left[\sum_{k=2}^{-1+n} \Sigma_{m[k]} \, (-x[-1+k]+x[k]) + \Sigma_i \, x[1]\right] V_j \, \text{xExpM}\left[\Sigma_j \, (d_{ij} - x[-1+n])\right]}{4\,\pi\, {d_{ij}}^2 \, (d_{ij} - x[-1+n])}$$

In particular:

```
In[ ]:= probabilityModel3 /. n → 2
```

$$\text{Out[ ]=}\quad P_{ij} \to \frac{\text{expM}\left[\Sigma_i \, x[1]\right] V_j \, \text{xExpM}\left[\Sigma_j \, (d_{ij} - x[1])\right]}{4\,\pi\, {d_{ij}}^2 \, (d_{ij} - x[1])}$$

Recover isotropic lossless scattering in the small optical thickness limit:

```
In[ ]:= probabilityModel3 /. {expM → (1 &), xExpM → Identity}
```

$$\text{Out[ ]=}\quad P_{ij} \to \frac{V_j \, \Sigma_j}{4\,\pi\, {d_{ij}}^2}$$

If the target region is thick, then the collision probability {15} *decreases* with optical thickness (due to the bumpy shape of xExpM, see section 1). However, as collision probability is a measure over space, it should *increase* with respect to optical thickness (at least in spherical or plane geometry where only one dimension of space is effective). Also, by self-shielding, the increase should be less than linear, which is indeed true with xExpM (although not strictly if x>1).

For consistency, any thick region should be decomposed in thinner regions. However, a cheaper solution is this patching:

Pij4

```
In[68]:= probabilityModel4 = probabilityModel3 /. xExpM → xExpM1
```

{16}

Pij4

$$\text{Out[68]=}\quad P_{ij} \to \left(\text{expM}\left[\sum_{k=2}^{-1+n} \Sigma_{m[k]} \, (-x[-1+k]+x[k]) + \Sigma_i \, x[1]\right] V_j \, \text{xExpM1}\left[\Sigma_j \, (d_{ij} - x[-1+n])\right]\right) \Big/ \left(4\,\pi\, {d_{ij}}^2 \, (d_{ij} - x[-1+n])\right)$$

```
In[ ]:= probabilityModel4 /. n → 2
```

$$\text{Out[ ]=}\quad P_{ij} \to \frac{\text{expM}\left[\Sigma_i \, x[1]\right] V_j \, \text{xExpM1}\left[\Sigma_j \, (d_{ij} - x[1])\right]}{4\,\pi\, {d_{ij}}^2 \, (d_{ij} - x[1])}$$

The patched probability model {16} has the consequence that $P_{ij} / (V_j \, \Sigma_j)$ in {16} is no more invariant by exchange, which breaks reciprocity {4}. As reciprocity is a consequence of the monokinetic hypothesis, the monokinetic hypothesis is broken too

but the collision probability model {16} may still have some meaning with respect to the non-monokinetic transport equation.

2spheres

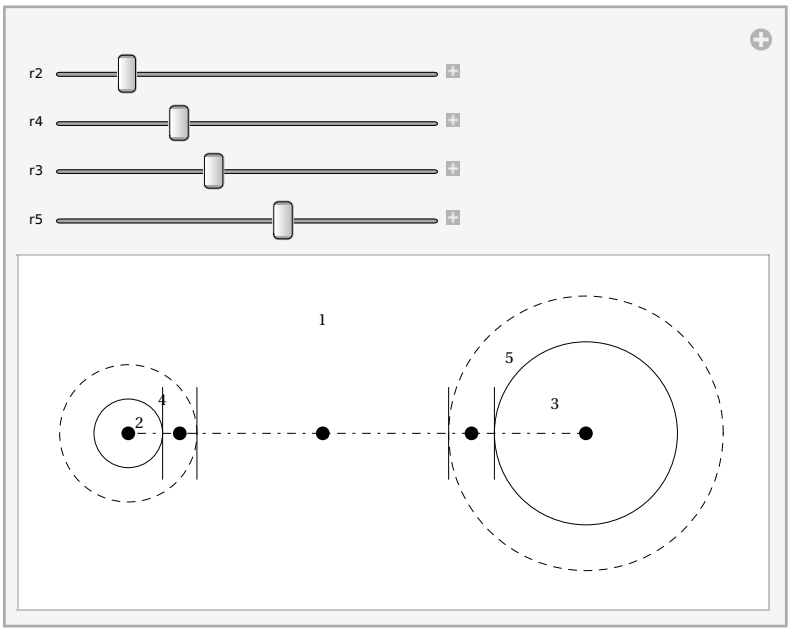
4.5. Two dressed spheres

Considering the roughness of the collision probability model, we cannot expect to account for region shape, except for the most global properties, volume and some mean or typical (optical) thickness, like the mean chord. At best, only sphere and slab can be considered.

For the small region hypothesis of the collision probability approximation, using only one region for the whole pile is the most patent violation. A little better can be done by considering around each small region an intermediary "dressing" region. The simplest choice for dressing is spherical corona, locally similar to slab. If the dressing material is the same as the pile material, then dressing thickness should have no effect. Thus can be checked consistency with respect to the small region hypothesis.

In[ ]:=
```
Catch @ Manipulate [With[{tick = .03 , d = 2},
    Graphics [#, Axes → {False , False }] &@{
      {DotDashed , Line @{{- d / 2 , 0}, {d / 2 , 0}}},
      Line @{{{#, #2}, {#, -#2}} & @@ {#, .2}} & /@ Table[(- d / 2 + x[i]), {i , 4}] /.
       {x[1] → r2 , x[2] → r4 , x[3] → d - r5 , x[4] → d - r3},
      {{Disk[#, tick], Text["1", # + {0 , .5}]} &@{Mean @{#1 , #2}, 0} & @@
          Composition [# /. {x_ , y_ , z_} :→ x - Sign @ x Max[y , z] &, MapAt[First , #, 1] &, Most] /@
           #,
        Through [
          {{Circle [#1 , #2], Text[ToString @#4 , #1 + {- Sign[First @#1], 1} #2 / 3],
              {Dashed , Circle [#1 , #3]}, Text[ToString [#4 + 2],
               #1 + {- Sign[First @#1], 1} (#2 + #3) / 3]} & @@ # &,
            {Disk[#1 , tick], Disk[#1 - Sign[First @#1] {(#2 + #3) / 2 , 0}, tick]} & @@ # &} @
           #] & /@ #} &@{{{- d / 2 , 0}, r2 , r4 , 2}, {{d / 2 , 0}, r3 , r5 , 3}}}],
  {{r2 , .15}, 0 , 1}, {{r4 , .3}, 0 , 1}, {{r3 , .4}, 0 , 1}, {{r5 , .6}, 0 , 1}]
```

Out[ ]=



In[69]:=
```
Clear @ indexMediaInterfaces ;
indexMediaInterfaces [media1_ , path1_ ] :=
 MapThread [Thread [Array [#1 , Length @ #2] → #2] &,
  {{m, x}, {media1 , Table [Sum[path1 [[i]], {i , 1, j}], {j , Length @ path1 }]}}];
```

Read pathwise on the figure and index regions and interfaces:

In[ ]:=
```
With[{
  media1 = {2, 4, 1, 5, 3},
  (*write so that Total would obviously return distance :*)
  path1 = {r2, - r2 + r4, - r4 + d23 - r5, r5 - r3, r3}},
 Print @ Total @ path1 ;
 indexMediaInterfaces [media1 , path1 ]
]
```

d_{23}

Out[ ]=
```
{{m[1] → 2 , m[2] → 4 , m[3] → 1 , m[4] → 5 , m[5] → 3},
 {x[1] → r2 , x[2] → r4 , x[3] → d23 - r5 , x[4] → d23 - r3 , x[5] → d23}}
```

Review formulae on the case:

```
In[ ]:= Column @ With[{media1 = {2, 4, 1, 5, 3}, path1 = {r2, -r2 + r4, -r4 + d23 - r5, r5 - r3, r3}},
  indexMediaInterfaces [media1, path1] //
   With[{
     media = First @ #,
     interfaces = Last @ #,
     n1 = Length @ media1,
     i1 = First @ media1,
     j1 = Last @ media1},
    With[{jmi = Join[media, interfaces]},
     {{media, interfaces},
      {n1, i1, j1},
      # /. {n → n1, i → i1, j → j1} /. jmi & /@
       {probabilityModel1, probabilityModel2, probabilityModel3, probabilityModel4},

      ReleaseHold [# /. {n → n1, i → i1, j → j1}] /. jmi & /@
       {opticalPath, opticalPath1, opticalPath2, opticalThickness},

      # /. {n → n1, i → i1, j → j1} /. jmi & /@ {opticalThickness1, opticalThickness2},

      # /. {n → n1, i → i1, j → j1} /. jmi /. {Σ4 → Σ1, Σ5 → Σ1} & @ opticalThickness2 //
       Simplify
     }]] &]
```

```
{{m[1] → 2, m[2] → 4, m[3] → 1, m[4] → 5, m[5] → 3},
 {x[1] → r2, x[2] → r4, x[3] → d23 - r5, x[4] → d23 - r3, x[5] → d23}}
{5, 2, 3}
{P23 → expM[(d23-r4-r5) Σ1+r2 Σ2+r3 Σ3+(-r2+r4) Σ4+(-r3+r5) Σ5] V3 Σ3 / (4 π d23^2),
 P23 → expM[(d23-r4-r5) Σ1+r2 Σ2+r3 Σ3+(-r2+r4) Σ4+(-r3+r5) Σ5] V3 Σ3 / (4 π d23^2),
 P23 → expM[(d23-r4-r5) Σ1+r2 Σ2+(-r2+r4) Σ4+(-r3+r5) Σ5] V3 xExpM[r3 Σ3] / (4 π d23^2 r3),
 P23 → expM[(d23-r4-r5) Σ1+r2 Σ2+(-r2+r4) Σ4+(-r3+r5) Σ5] V3 xExpM1[r3 Σ3] / (4 π d23^2 r3)}
```

Out[ ]=

```
{{r2 Σ2, (-r2 + r4) Σ4, (d23 - r4 - r5) Σ1, (-r3 + r5) Σ5, r3 Σ3},
 {r2 Σ2, (-r2 + r4) Σ4, (d23 - r4 - r5) Σ1, (-r3 + r5) Σ5, r3 Σ3},
 {r2 Σ2, (-r2 + r4) Σ4, (d23 - r4 - r5) Σ1, (-r3 + r5) Σ5, r3 Σ3},
 (d23 - r4 - r5) Σ1 + r2 Σ2 + r3 Σ3 + (-r2 + r4) Σ4 + (-r3 + r5) Σ5}
{(d23 - r4 - r5) Σ1 + r2 Σ2 + r3 Σ3 + (-r2 + r4) Σ4 + (-r3 + r5) Σ5,
 (d23 - r4 - r5) Σ1 + r2 Σ2 + r3 Σ3 + (-r2 + r4) Σ4 + (-r3 + r5) Σ5}
d23 Σ1 + r2 (-Σ1 + Σ2) + r3 (-Σ1 + Σ3)
```

Collapse points (the black dots on the figure) are chosen for (local) symmetry at sphere centers or at the middle of linear sections. All collision probabilities between any two regions are obtained by recursively collapsing extremal media (in both directions):

In[70]:=
```
Clear @ collapse ;
collapse [side_ : (First | Last)] @ {media_List , path_List } :=
 With [{order = Switch [side , First , Identity , Last , Reverse ]},
  {order [order @ media /. {_, m2_ , m3__ } :> {m2 , m3}],
   order [order @ path /. {_, d2_ , d3__ } :> {d2 / 2 , d3}]}]
```

In[ ]:=
```
collapse [#] @ {{2, 4, 1, 5, 3}, {r2, -r2 + r4, d23 - r4 - r5, -r3 + r5, r3}} & /@ {First , Last}
```

Out[ ]=
$$\left\{\left\{\{4,\ 1,\ 5,\ 3\},\ \left\{\frac{1}{2}(-r_2+r_4),\ d_{23}-r_4-r_5,\ -r_3+r_5,\ r_3\right\}\right\},\ \left\{\{2,\ 4,\ 1,\ 5\},\ \left\{r_2,\ -r_2+r_4,\ d_{23}-r_4-r_5,\ \frac{1}{2}(-r_3+r_5)\right\}\right\}\right\}$$

In[ ]:=
```
TableForm [Most @ FixedPointList [collapse @ #, {Range @ #, Array [d, {#}]} & @ 5] & /@
  {First , Last}, TableSpacing → {2, 3}]
```

Out[ ]//TableForm=

1	2	3	4	5	2	3	4	5	3	4	5	4	5	
d[1]	d[2]	d[3]	d[4]	d[5]	$\frac{d[2]}{2}$	d[3]	d[4]	d[5]	$\frac{d[3]}{2}$	d[4]	d[5]	$\frac{d[4]}{2}$	d[5]	
1	2	3	4	5	1	2	3	4	1	2	3	1	2	
d[1]	d[2]	d[3]	d[4]	d[5]	d[1]	d[2]	d[3]	$\frac{d[4]}{2}$	d[1]	d[2]	$\frac{d[3]}{2}$	d[1]	$\frac{d[2]}{2}$	

In[ ]:=
```
Most @ FixedPointList [collapse @ First ,
   {{2, 4, 1, 5, 3}, {r2, -r2 + r4, d23 - r4 - r5, -r3 + r5, r3}}] // Column
```

Out[ ]=
$$\{\{2,\ 4,\ 1,\ 5,\ 3\},\ \{r_2,\ -r_2+r_4,\ d_{23}-r_4-r_5,\ -r_3+r_5,\ r_3\}\}$$
$$\left\{\{4,\ 1,\ 5,\ 3\},\ \left\{\frac{1}{2}(-r_2+r_4),\ d_{23}-r_4-r_5,\ -r_3+r_5,\ r_3\right\}\right\}$$
$$\left\{\{1,\ 5,\ 3\},\ \left\{\frac{1}{2}(d_{23}-r_4-r_5),\ -r_3+r_5,\ r_3\right\}\right\}$$
$$\left\{\{5,\ 3\},\ \left\{\frac{1}{2}(-r_3+r_5),\ r_3\right\}\right\}$$

In[ ]:=
```
Flatten [Most @ FixedPointList [collapse @ Last , #] & /@
   Most @ FixedPointList [collapse @ First ,
     {{2, 4, 1, 5, 3}, {r2, -r2 + r4, d23 - r4 - r5, -r3 + r5, r3}}], 1] // Column
```

Out[ ]=
$$\{\{2,\ 4,\ 1,\ 5,\ 3\},\ \{r_2,\ -r_2+r_4,\ d_{23}-r_4-r_5,\ -r_3+r_5,\ r_3\}\}$$
$$\left\{\{2,\ 4,\ 1,\ 5\},\ \left\{r_2,\ -r_2+r_4,\ d_{23}-r_4-r_5,\ \frac{1}{2}(-r_3+r_5)\right\}\right\}$$
$$\left\{\{2,\ 4,\ 1\},\ \left\{r_2,\ -r_2+r_4,\ \frac{1}{2}(d_{23}-r_4-r_5)\right\}\right\}$$
$$\left\{\{2,\ 4\},\ \left\{r_2,\ \frac{1}{2}(-r_2+r_4)\right\}\right\}$$
$$\left\{\{4,\ 1,\ 5,\ 3\},\ \left\{\frac{1}{2}(-r_2+r_4),\ d_{23}-r_4-r_5,\ -r_3+r_5,\ r_3\right\}\right\}$$
$$\left\{\{4,\ 1,\ 5\},\ \left\{\frac{1}{2}(-r_2+r_4),\ d_{23}-r_4-r_5,\ \frac{1}{2}(-r_3+r_5)\right\}\right\}$$
$$\left\{\{4,\ 1\},\ \left\{\frac{1}{2}(-r_2+r_4),\ \frac{1}{2}(d_{23}-r_4-r_5)\right\}\right\}$$
$$\left\{\{1,\ 5,\ 3\},\ \left\{\frac{1}{2}(d_{23}-r_4-r_5),\ -r_3+r_5,\ r_3\right\}\right\}$$
$$\left\{\{1,\ 5\},\ \left\{\frac{1}{2}(d_{23}-r_4-r_5),\ \frac{1}{2}(-r_3+r_5)\right\}\right\}$$
$$\left\{\{5,\ 3\},\ \left\{\frac{1}{2}(-r_3+r_5),\ r_3\right\}\right\}$$

In[ ]:=
```
indexMediaInterfaces [{2, 4, 1, 5, 3}, {r2, -r2 + r4, d23 - r4 - r5, -r3 + r5, r3}]
```

Out[ ]=
$$\{\{m[1] \to 2,\ m[2] \to 4,\ m[3] \to 1,\ m[4] \to 5,\ m[5] \to 3\},\ \{x[1] \to r_2,\ x[2] \to r_4,\ x[3] \to d_{23} - r_5,\ x[4] \to d_{23} - r_3,\ x[5] \to d_{23}\}\}$$

In[ ]:=
```
indexMediaInterfaces @@ # & /@
  Flatten[Most @ FixedPointList [collapse @ Last, #] & /@
    Most @ FixedPointList [collapse @ First,
      {{2, 4, 1, 5, 3}, {r2, -r2 + r4, d23 - r4 - r5, -r3 + r5, r3}}], 1] // Identity //
 TableForm [#, TableDepth → 2, TableDirections → {Column, Column}, TableSpacing → {2, 1}] &
```

Out[ ]//TableForm=

$\{m[1] \to 2,\ m[2] \to 4,\ m[3] \to 1,\ m[4] \to 5,\ m[5] \to 3\}$
$\{x[1] \to r_2,\ x[2] \to r_4,\ x[3] \to d_{23} - r_5,\ x[4] \to d_{23} - r_3,\ x[5] \to d_{23}\}$

$\{m[1] \to 2,\ m[2] \to 4,\ m[3] \to 1,\ m[4] \to 5\}$
$\left\{x[1] \to r_2,\ x[2] \to r_4,\ x[3] \to d_{23} - r_5,\ x[4] \to d_{23} - r_5 + \frac{1}{2}(-r_3 + r_5)\right\}$

$\{m[1] \to 2,\ m[2] \to 4,\ m[3] \to 1\}$
$\left\{x[1] \to r_2,\ x[2] \to r_4,\ x[3] \to r_4 + \frac{1}{2}(d_{23} - r_4 - r_5)\right\}$

$\{m[1] \to 2,\ m[2] \to 4\}$
$\left\{x[1] \to r_2,\ x[2] \to r_2 + \frac{1}{2}(-r_2 + r_4)\right\}$

$\{m[1] \to 4,\ m[2] \to 1,\ m[3] \to 5,\ m[4] \to 3\}$
$\left\{x[1] \to \frac{1}{2}(-r_2 + r_4),\ x[2] \to d_{23} - r_4 + \frac{1}{2}(-r_2 + r_4) - r_5,\ x[3] \to d_{23} - r_3 - r_4 + \frac{1}{2}(-r_2 + r_4),\ x[4] \to d_{23} - r_4 + \frac{1}{2}\right.$

$\{m[1] \to 4,\ m[2] \to 1,\ m[3] \to 5\}$
$\left\{x[1] \to \frac{1}{2}(-r_2 + r_4),\ x[2] \to d_{23} - r_4 + \frac{1}{2}(-r_2 + r_4) - r_5,\ x[3] \to d_{23} - r_4 + \frac{1}{2}(-r_2 + r_4) - r_5 + \frac{1}{2}(-r_3 + r_5)\right\}$

$\{m[1] \to 4,\ m[2] \to 1\}$
$\left\{x[1] \to \frac{1}{2}(-r_2 + r_4),\ x[2] \to \frac{1}{2}(-r_2 + r_4) + \frac{1}{2}(d_{23} - r_4 - r_5)\right\}$

$\{m[1] \to 1,\ m[2] \to 5,\ m[3] \to 3\}$
$\left\{x[1] \to \frac{1}{2}(d_{23} - r_4 - r_5),\ x[2] \to -r_3 + \frac{1}{2}(d_{23} - r_4 - r_5) + r_5,\ x[3] \to \frac{1}{2}(d_{23} - r_4 - r_5) + r_5\right\}$

$\{m[1] \to 1,\ m[2] \to 5\}$
$\left\{x[1] \to \frac{1}{2}(d_{23} - r_4 - r_5),\ x[2] \to \frac{1}{2}(d_{23} - r_4 - r_5) + \frac{1}{2}(-r_3 + r_5)\right\}$

$\{m[1] \to 5,\ m[2] \to 3\}$
$\left\{x[1] \to \frac{1}{2}(-r_3 + r_5),\ x[2] \to r_3 + \frac{1}{2}(-r_3 + r_5)\right\}$

In[72]:=
```
expandProbability  = Function [{probabilityModel  , media1 , path1 },
    Composition [
      Fold[ReplaceAll , probabilityModel  , #] &,
      {Sequence  @@ With[{n1 = Length @ #1 , i1 = m[1] /. #1},
             With[{j1 = m[n1] /. #1},
              {{n → n1 , i → i1 , j → j1}, {d_{i1 j1} → (x[n1] /. #2)}}]],
           Join[#1 , Most @ #2] & @@ # &,
      indexMediaInterfaces  @@ # &] /@
     Flatten [{#, Map[Reverse , #, {2}]} &@
       Flatten [Most @ FixedPointList [collapse @ Last , #] & /@ Most @
           FixedPointList [
            collapse @ First , {media1 , path1 }], 1], 1]];
```

```
expandProbability [probabilityModel4  , {2, 4, 1, 5, 3}, {r_2, -r_2 + r_4, -r_4 + d_{23} - r_5, r_5 - r_3, r_3}]
```

Out[ ]=

$$\left\{P_{23} \to \frac{1}{4\,\pi\,{d_{23}}^2\,r_3}\,\text{expM}\left[(d_{23}-r_4-r_5)\,\Sigma_1+r_2\,\Sigma_2+(-r_2+r_4)\,\Sigma_4+(-r_3+r_5)\,\Sigma_5\right]\,V_3\,\text{xExpM1}\left[r_3\,\Sigma_3\right],\right.$$

$$P_{25} \to \frac{\text{expM}\left[(d_{23}-r_4-r_5)\,\Sigma_1+r_2\,\Sigma_2+(-r_2+r_4)\,\Sigma_4\right]\,V_5\,\text{xExpM1}\left[\frac{1}{2}\,(-r_3+r_5)\,\Sigma_5\right]}{2\,\pi\,(-r_3+r_5)\left(d_{23}-r_5+\frac{1}{2}\,(-r_3+r_5)\right)^2},$$

$$P_{21} \to \frac{\text{expM}\left[r_2\,\Sigma_2+(-r_2+r_4)\,\Sigma_4\right]\,V_1\,\text{xExpM1}\left[\frac{1}{2}\,(d_{23}-r_4-r_5)\,\Sigma_1\right]}{2\,\pi\left(r_4+\frac{1}{2}\,(d_{23}-r_4-r_5)\right)^2\,(d_{23}-r_4-r_5)},$$

$$P_{24} \to \frac{\text{expM}\left[r_2\,\Sigma_2\right]\,V_4\,\text{xExpM1}\left[\frac{1}{2}\,(-r_2+r_4)\,\Sigma_4\right]}{2\,\pi\,(-r_2+r_4)\left(r_2+\frac{1}{2}\,(-r_2+r_4)\right)^2},$$

$$P_{43} \to \left(\text{expM}\left[\left(d_{23}+\frac{1}{2}\,(r_2-r_4)-r_4+\frac{1}{2}\,(-r_2+r_4)-r_5\right)\Sigma_1+\frac{1}{2}\,(-r_2+r_4)\,\Sigma_4+\right.\right.$$
$$\left.\left.\left(-r_3+\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)+r_5\right)\Sigma_5\right]\,V_3\,\text{xExpM1}\left[\left(r_3+\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)\right)\Sigma_3\right]\right)\Big/$$
$$\left(4\,\pi\left(r_3+\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)\right)\left(d_{23}-r_4+\frac{1}{2}\,(-r_2+r_4)\right)^2\right),$$

$$P_{45} \to \left(\text{expM}\left[\left(d_{23}+\frac{1}{2}\,(r_2-r_4)-r_4+\frac{1}{2}\,(-r_2+r_4)-r_5\right)\Sigma_1+\frac{1}{2}\,(-r_2+r_4)\,\Sigma_4\right]\right.$$
$$\left.V_5\,\text{xExpM1}\left[\left(\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)+\frac{1}{2}\,(-r_3+r_5)\right)\Sigma_5\right]\right)\Big/$$
$$\left(4\,\pi\left(\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)+\frac{1}{2}\,(-r_3+r_5)\right)\left(d_{23}-r_4+\frac{1}{2}\,(-r_2+r_4)-r_5+\frac{1}{2}\,(-r_3+r_5)\right)^2\right),$$

$$P_{41} \to \left(\text{expM}\left[\frac{1}{2}\,(-r_2+r_4)\,\Sigma_4\right]\,V_1\,\text{xExpM1}\left[\left(\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)+\frac{1}{2}\,(d_{23}-r_4-r_5)\right)\Sigma_1\right]\right)\Big/$$
$$\left(4\,\pi\left(\frac{1}{2}\,(-r_2+r_4)+\frac{1}{2}\,(d_{23}-r_4-r_5)\right)^2\left(\frac{1}{2}\,(r_2-r_4)+\frac{1}{2}\,(-r_2+r_4)+\frac{1}{2}\,(d_{23}-r_4-r_5)\right)\right),$$

$$P_{13} \to \left(\text{expM}\left[\frac{1}{2}\,(d_{23}-r_4-r_5)\,\Sigma_1+\left(-r_3+\frac{1}{2}\,(d_{23}-r_4-r_5)+r_5+\frac{1}{2}\,(-d_{23}+r_4+r_5)\right)\Sigma_5\right]\right.$$

$$V_3\ \text{xExpM1}\left[\left(r_3+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\Sigma_3\right]\Big/$$
$$\left(4\pi\left(\frac{1}{2}(d_{23}-r_4-r_5)+r_5\right)^2\left(r_3+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\right),$$

$$P_{15}\to\left(\text{expM}\left[\frac{1}{2}(d_{23}-r_4-r_5)\Sigma_1\right]V_5\ \text{xExpM1}\left[\left(\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\Sigma_5\right]\right)\Big/$$
$$\left(4\pi\left(\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)\right)^2\left(\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\right),$$

$$P_{53}\to\frac{\text{expM}\left[\frac{1}{2}(-r_3+r_5)\Sigma_5\right]V_3\ \text{xExpM1}\left[\left(r_3+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\Sigma_3\right]}{4\pi\left(r_3+\frac{1}{2}(-r_3+r_5)\right)^2\left(r_3+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)},$$

$$P_{32}\to\frac{1}{4\pi\,{d_{23}}^2\,r_2}\,\text{expM}\left[(d_{23}-r_4-r_5)\Sigma_1+r_3\Sigma_3+(-r_2+r_4)\Sigma_4+(-r_3+r_5)\Sigma_5\right]V_2\ \text{xExpM1}\left[r_2\Sigma_2\right],$$

$$P_{52}\to\left(\text{expM}\left[\left(d_{23}-r_4+\frac{1}{2}(r_3-r_5)-r_5+\frac{1}{2}(-r_3+r_5)\right)\Sigma_1+\left(-r_2+r_4+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\Sigma_4+\right.\right.$$
$$\left.\left.\frac{1}{2}(-r_3+r_5)\Sigma_5\right]V_2\ \text{xExpM1}\left[\left(r_2+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\Sigma_2\right]\right)\Big/$$
$$\left(4\pi\left(r_2+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\left(d_{23}-r_5+\frac{1}{2}(-r_3+r_5)\right)^2\right),$$

$$P_{12}\to\left(\text{expM}\left[\frac{1}{2}(d_{23}-r_4-r_5)\Sigma_1+\left(-r_2+r_4+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\Sigma_4\right]\right.$$
$$\left.V_2\ \text{xExpM1}\left[\left(r_2+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\Sigma_2\right]\right)\Big/$$
$$\left(4\pi\left(r_4+\frac{1}{2}(d_{23}-r_4-r_5)\right)^2\left(r_2+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\right),$$

$$P_{42}\to\frac{\text{expM}\left[\frac{1}{2}(-r_2+r_4)\Sigma_4\right]V_2\ \text{xExpM1}\left[\left(r_2+\frac{1}{2}(r_2-r_4)+\frac{1}{2}(-r_2+r_4)\right)\Sigma_2\right]}{4\pi\left(r_2+\frac{1}{2}(-r_2+r_4)\right)^2\left(r_2+\frac{1}{2}(r_2-r_4)+\frac{1}{2}(-r_2+r_4)\right)},$$

$$P_{34}\to\frac{\text{expM}\left[(d_{23}-r_4-r_5)\Sigma_1+r_3\Sigma_3+(-r_3+r_5)\Sigma_5\right]V_4\ \text{xExpM1}\left[\frac{1}{2}(-r_2+r_4)\Sigma_4\right]}{2\pi(-r_2+r_4)\left(d_{23}-r_4+\frac{1}{2}(-r_2+r_4)\right)^2},$$

$$P_{54}\to\left(\text{expM}\left[\left(d_{23}-r_4+\frac{1}{2}(r_3-r_5)-r_5+\frac{1}{2}(-r_3+r_5)\right)\Sigma_1+\frac{1}{2}(-r_3+r_5)\Sigma_5\right]\right.$$
$$\left.V_4\ \text{xExpM1}\left[\left(\frac{1}{2}(-r_2+r_4)+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\Sigma_4\right]\right)\Big/$$
$$\left(4\pi\left(\frac{1}{2}(-r_2+r_4)+\frac{1}{2}(r_3-r_5)+\frac{1}{2}(-r_3+r_5)\right)\left(d_{23}-r_4+\frac{1}{2}(-r_2+r_4)-r_5+\frac{1}{2}(-r_3+r_5)\right)^2\right),$$

$$P_{14}\to\left(\text{expM}\left[\frac{1}{2}(d_{23}-r_4-r_5)\Sigma_1\right]V_4\ \text{xExpM1}\left[\left(\frac{1}{2}(-r_2+r_4)+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\Sigma_4\right]\right)\Big/$$
$$\left(4\pi\left(\frac{1}{2}(-r_2+r_4)+\frac{1}{2}(d_{23}-r_4-r_5)\right)^2\left(\frac{1}{2}(-r_2+r_4)+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-d_{23}+r_4+r_5)\right)\right),$$

$$P_{31}\to\frac{\text{expM}\left[r_3\Sigma_3+(-r_3+r_5)\Sigma_5\right]V_1\ \text{xExpM1}\left[\frac{1}{2}(d_{23}-r_4-r_5)\Sigma_1\right]}{2\pi(d_{23}-r_4-r_5)\left(\frac{1}{2}(d_{23}-r_4-r_5)+r_5\right)^2},$$

$$P_{51} \to \left(\text{expM}\left[\frac{1}{2}(-r_3+r_5)\,\Sigma_5\right] V_1\ \text{xExpM1}\left[\left(\frac{1}{2}(r_3-r_5)+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)\right)\Sigma_1\right]\right)\Big/$$
$$\left(4\,\pi\left(\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)\right)^2\left(\frac{1}{2}(r_3-r_5)+\frac{1}{2}(d_{23}-r_4-r_5)+\frac{1}{2}(-r_3+r_5)\right)\right),$$

$$P_{35} \to \frac{\text{expM}\left[r_3\,\Sigma_3\right] V_5\ \text{xExpM1}\left[\frac{1}{2}(-r_3+r_5)\,\Sigma_5\right]}{2\,\pi\,(-r_3+r_5)\left(r_3+\frac{1}{2}(-r_3+r_5)\right)^2}\Big\}$$

In[ ]:=
```
Length @ %
```

Out[ ]=
```
20
```

4spheres

4.6. Four spheres

They can have any position or size inside the larger spherical pile, although shadow and boundary effects are not accounted for. The two dressed sphere model is applied to each pair of spheres separately. The region indexing is modified so that if dressing regions were removed then the stochastic matrix would shrink to a single full block. Example with 4 spheres:

```
Length /@ Function [{probabilityModel , n0},
   Simplify [
    Function [{i, j}, expandProbability [probabilityModel , {i, i + n0, 1, j + n0, j},
         {ri, -ri + ri+n0, -ri+n0 + dij - rj+n0, rj+n0 - rj, rj}]] @@ # & /@
     Subsets [Range @ n0 + 1, {2}]]][probabilityModel4 , 4]
```

Out[ ]=
```
{20, 20, 20, 20, 20, 20}
```

The P_{1i} and $P_{i,i+4}$, $P_{i+4,i}$, $2 \le i \le 5$ and are evaluated thrice, depending on the axis. Yet for the $P_{i,i+4}$, $P_{i+4,i}$ the three values are identical by spherical symmetry:

```
Sort @ Cases [Function [{probabilityModel , n0},
     Composition [
       Flatten [#, 1] &]@
      Simplify [
       Function [{i, j}, expandProbability [probabilityModel , {i, i + n0, 1, j + n0, j},
            {r_i, -r_i + r_(i+n0), -r_(i+n0) + d_ij - r_(j+n0), r_(j+n0) - r_j, r_j}]] @@ # & /@
        Subsets [Range @ n0 + 1, {2}]]][probabilityModel4 , 4], (P_26 → _) | (P_12 → _)]
```

Out[ ]=

$$\left\{P_{12} \to \frac{\text{expM}\left[\frac{1}{2}\left(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6\right)\right] V_2\ \text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\, r_2\,(d_{23} + r_6 - r_7)^2},\right.$$

$$P_{12} \to \frac{\text{expM}\left[\frac{1}{2}\left(d_{24}\,\Sigma_1 - r_8\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6\right)\right] V_2\ \text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\, r_2\,(d_{24} + r_6 - r_8)^2},$$

$$P_{12} \to \frac{\text{expM}\left[\frac{1}{2}\left(d_{25}\,\Sigma_1 - r_9\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6\right)\right] V_2\ \text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\, r_2\,(d_{25} + r_6 - r_9)^2},$$

$$P_{26} \to -\frac{2\ \text{expM}\left[r_2\,\Sigma_2\right] V_6\ \text{xExpM1}\left[\frac{1}{2}\left(-r_2 + r_6\right)\Sigma_6\right]}{\pi\,(r_2 - r_6)\,(r_2 + r_6)^2},$$

$$\left.P_{26} \to -\frac{2\ \text{expM}\left[r_2\,\Sigma_2\right] V_6\ \text{xExpM1}\left[\frac{1}{2}\left(-r_2 + r_6\right)\Sigma_6\right]}{\pi\,(r_2 - r_6)\,(r_2 + r_6)^2},\ P_{26} \to -\frac{2\ \text{expM}\left[r_2\,\Sigma_2\right] V_6\ \text{xExpM1}\left[\frac{1}{2}\left(-r_2 + r_6\right)\Sigma_6\right]}{\pi\,(r_2 - r_6)\,(r_2 + r_6)^2}\right\}$$

```
Function [{probabilityModel , n0},
  Composition [
    #[[1, 1]] & /@ # &,
    Gather [#, Map[First , #1 == #2 &, {2}]] &,
    Union ,
    Flatten [#, 1] &]@
   Simplify [
    Function [{i, j}, expandProbability [probabilityModel , {i, i + n0, 1, j + n0, j},
         {r_i, -r_i + r_(i+n0), -r_(i+n0) + d_ij - r_(j+n0), r_(j+n0) - r_j, r_j}]] @@ # & /@
     Subsets [Range @ n0 + 1, {2}]]][probabilityModel4 , 4]
```

Out[ ]=

$$\{P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{19}, P_{21}, P_{23}, P_{24}, P_{25}, P_{26}, P_{27}, P_{28}, P_{29}, P_{31}, P_{32},$$
$$P_{34}, P_{35}, P_{36}, P_{37}, P_{38}, P_{39}, P_{41}, P_{42}, P_{43}, P_{45}, P_{46}, P_{47}, P_{48}, P_{49}, P_{51}, P_{52}, P_{53}, P_{54},$$
$$P_{56}, P_{57}, P_{58}, P_{59}, P_{61}, P_{62}, P_{63}, P_{64}, P_{65}, P_{67}, P_{68}, P_{69}, P_{71}, P_{72}, P_{73}, P_{74}, P_{75}, P_{76},$$
$$P_{78}, P_{79}, P_{81}, P_{82}, P_{83}, P_{84}, P_{85}, P_{86}, P_{87}, P_{89}, P_{91}, P_{92}, P_{93}, P_{94}, P_{95}, P_{96}, P_{97}, P_{98}\}$$

As the 9 diagonal collision probabilities are missing, we obtain as many rules as all (i , j), i ≠ j:

In[ ]:=
```
Length @ %
(*9*9-9==72*)
```

Out[ ]= 72

The anisotropic error will be estimated by mean deviation:

In[304]:=
```
MeanDeviation [{p1, p2, p3}]
```

Out[304]=
$$\frac{1}{3}\left(\text{Abs}\left[\text{p1}+\frac{1}{3}(-\text{p1}-\text{p2}-\text{p3})\right]+\text{Abs}\left[\text{p2}+\frac{1}{3}(-\text{p1}-\text{p2}-\text{p3})\right]+\text{Abs}\left[\frac{1}{3}(-\text{p1}-\text{p2}-\text{p3})+\text{p3}\right]\right)$$

The average of collision probabilities on three different axes is loosely related to the integral on all possible paths of the exact method.

In[73]:=
```
Clear @ expandProbability1 ;
expandProbability1 [probabilityModel_ , (*number of spheres *)n0_Integer ,
  OptionsPattern [dressing → True]] := Catch @ Composition [
    Partition [#, If[OptionValue @ dressing , 2, 1] n0] &,
    #[[1, 1]] → Through [{(*hangs : Simplify @**)Mean ,
             MeanDeviation @ #
                  Mean @ # &}[Last /@ #]] & /@ # &,
    Gather [#, Map[First , #1 == #2 &, {2}]] &,
    Union ,
    Flatten [#, 1] &]@
   Simplify [
    Function [{i, j}, expandProbability [probabilityModel , If[OptionValue @ dressing ,
          (*this could be determined automatically from a more detailed
           geometrical description and shadows could be found too*)
          {i, i + n0, 1, j + n0, j}, {i, 1, j}],
         (*not a good idea*)
         (*MapAt [
            If[False ,Max[#,0]&,Identity ],
            #,
            (*middle of list of uneven length *)
            1+Round [Length @#/2]]&@*)If[OptionValue @ dressing ,
          {r_i, -r_i + r_(i+n0), -r_(i+n0) + d_ij - r_(j+n0), r_(j+n0) - r_j, r_j},
          {r_i, -r_i + d_ij - r_j, r_j}]
        ]] @@ # & /@ Subsets [Range @ n0 + 1, {2}]]
```

In[ ]:=
```
expandProbability2  = expandProbability1 [probabilityModel4 , 4];
```

In[ ]:=
```
ByteCount @ expandProbability2
```

Out[ ]=
```
1 132 504
```

In[ ]:=
```
expandProbability2 [[2, 4]]
```

Out[ ]=
$$\text{P}_{25} \to \left\{\frac{1}{4\,\pi\,\text{d}_{25}{}^2\,\text{r}_5}\,\text{expM}\left[(\text{d}_{25}-\text{r}_6-\text{r}_9)\,\Sigma_1+\text{r}_2\,\Sigma_2+(-\text{r}_2+\text{r}_6)\,\Sigma_6+(-\text{r}_5+\text{r}_9)\,\Sigma_9\right]\text{V}_5\,\text{xExpM1}\left[\text{r}_5\,\Sigma_5\right],\,0\right\}$$

```
In[ ]:= expandProbability2 [[4 + 1, 2]]
```

$$\text{Out[ ]= } P_{52} \to \left\{ \frac{1}{4\,\pi\,{d_{25}}^2\,r_2}\,\text{expM}\left[(d_{25} - r_6 - r_9)\,\Sigma_1 + r_5\,\Sigma_5 + (-r_2 + r_6)\,\Sigma_6 + (-r_5 + r_9)\,\Sigma_9\right] V_2\,\text{xExpM1}\left[r_2\,\Sigma_2\right],\ 0 \right\}$$

As union also sorts, partitioning yields a square matrix, less the diagonal:

```
In[ ]:= MatrixForm @ Map[First , expandProbability2  , {2}]
```

$$\text{Out[ ]//MatrixForm=} \begin{pmatrix} P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} & P_{19} \\ P_{21} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} & P_{28} & P_{29} \\ P_{31} & P_{32} & P_{34} & P_{35} & P_{36} & P_{37} & P_{38} & P_{39} \\ P_{41} & P_{42} & P_{43} & P_{45} & P_{46} & P_{47} & P_{48} & P_{49} \\ P_{51} & P_{52} & P_{53} & P_{54} & P_{56} & P_{57} & P_{58} & P_{59} \\ P_{61} & P_{62} & P_{63} & P_{64} & P_{65} & P_{67} & P_{68} & P_{69} \\ P_{71} & P_{72} & P_{73} & P_{74} & P_{75} & P_{76} & P_{78} & P_{79} \\ P_{81} & P_{82} & P_{83} & P_{84} & P_{85} & P_{86} & P_{87} & P_{89} \\ P_{91} & P_{92} & P_{93} & P_{94} & P_{95} & P_{96} & P_{97} & P_{98} \end{pmatrix}$$

```
In[ ]:= Dimensions @ %
```

```
Out[ ]= {9, 8}
```

```
In[ ]:= MatrixForm @ Map[First , expandProbability1 [probabilityModel4  , 4, dressing → False], {2}]
```

$$\text{Out[ ]//MatrixForm=} \begin{pmatrix} P_{12} & P_{13} & P_{14} & P_{15} \\ P_{21} & P_{23} & P_{24} & P_{25} \\ P_{31} & P_{32} & P_{34} & P_{35} \\ P_{41} & P_{42} & P_{43} & P_{45} \\ P_{51} & P_{52} & P_{53} & P_{54} \end{pmatrix}$$

Anisotropy (anisotropic error) α_{ij}:

```
In[ ]:= anisotropy1  = Map[# /. (P_i_j_ → {_, meanDeviation_ }) :> α_ij → meanDeviation  &,
          expandProbability2  , {2}];
```

```
In[ ]:= anisotropy1 [[1, 1]] // Short[#, 9] &
```

$$\text{Out[ ]//Short=} \quad \alpha_{12} \to \left(\text{Abs}\left[\frac{\text{expM}\left[\frac{1}{2}\,(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6)\right] V_2\,\text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\,r_2\,(d_{23} + r_6 - r_7)^2} + \frac{1}{3} \right.\right.$$
$$\left(-\left(\left(\text{expM}\left[\frac{1}{2}\,(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6)\right] V_2 \ll 1 \gg \text{xExpM1}\left[\ll 1 \gg\right] \right) \Big/ \left(\pi\,r_2\,(d_{23} + r_6 - r_7)^2\right) \right) -$$
$$\left.\left. \frac{\ll 1 \gg}{\pi\,r_2 \ll 1 \gg^2} - \frac{\ll 1 \gg}{\pi\,r_2\,(\ll 1 \gg)^2} \right) \right] + \ll 1 \gg + \text{Abs}\left[\frac{\ll 1 \gg}{\ll 1 \gg} + \frac{1}{3}\,(\ll 1 \gg) \right] \Bigg) \Big/$$
$$\left(\frac{\text{expM}\left[\frac{1}{2}\,(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6)\right] V_2\,\text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\,r_2\,(d_{23} + r_6 - r_7)^2} + \frac{\ll 1 \gg}{\pi\,r_2\,(\ll 1 \gg)^2} + \right.$$
$$\left. \frac{\text{expM}\left[\frac{1}{2}\,(d_{25}\,\Sigma_1 - r_9\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6)\right] V_2\,\text{xExpM1}\left[r_2\,\Sigma_2\right]}{\pi\,r_2\,(d_{25} + r_6 - r_9)^2} \right)$$

```
In[ ]:= With[{anisotropy2 = MapThread[#1 /. #2 &, {Table[α_ij, {i, 9}, {j, 9}], anisotropy1}]},
  Print@anisotropy2[[1, 1]];
  anisotropy2[[1, 2]] // Short[#, 8] &
 ]
```

α_{11}

Out[ ]//Short=

$$\left(\text{Abs}\left[\frac{\text{expM}\left[\frac{1}{2}\left(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6\right)\right] V_2\,\text{xExpM1}[r_2\,\Sigma_2]}{\pi\,r_2\,(d_{23} + r_6 - r_7)^2} + \right.\right.$$

$$\left.\frac{1}{3}\left(-\frac{\text{expM}\left[\frac{1}{2}\,(\ll 1\gg)\right] V_2\,\text{xExpM1}[r_2\,\Sigma_2]}{\pi\,r_2\,(d_{23} + r_6 - r_7)^2} - \frac{\text{expM}[\ll 1\gg]\,V_2 \ll 1\gg}{\pi\,r_2\,(\ll 1\gg)^2} - \frac{\ll 1\gg}{\pi\,r_2\,(\ll 1\gg)^2}\right)\right] + \ll 1\gg +$$

$$\left.\text{Abs}\left[\frac{\ll 1\gg}{\ll 1\gg} + \ll 1\gg\right]\right) \Big/ \left(\frac{\text{expM}\left[\frac{1}{2}\left(d_{23}\,\Sigma_1 - r_7\,\Sigma_1 - r_6\,(\Sigma_1 - 2\,\Sigma_6) - 2\,r_2\,\Sigma_6\right)\right] V_2\,\text{xExpM1}[r_2\,\Sigma_2]}{\pi\,r_2\,(d_{23} + r_6 - r_7)^2} + \right.$$

$$\left.\left.\frac{\text{expM}\left[\frac{1}{2}\,(\ll 1\gg)\right] V_2\,\text{xExpM1}[r_2\,\Sigma_2]}{\pi\,r_2\,(d_{24} + r_6 - r_8)^2} + \frac{\ll 1\gg}{\pi\,r_2\,(d_{25} + r_6 - r_9)^2}\right)\right.$$

indirectCollision

5. Indirect collision experiment

An extraneous source emits particles that collide with a sample and the products are detected, just as in any particle physics experiments, except that the medium is not vacuum but a scattering medium: the *pile*. Multiplicative scattering is when one incident particle yields many scattered particles (not to be confused with multiple scattering, when one particle is scattered many times).

The experiment needs five more or less localized functions:
1 - Pile,
2 - Sample,
3 - Detector,
4 - Extraneous source,
5 - Control system.

The detector output is a collision number observed in a given time interval (a collision rate). Nothing else can normally be observed (by definition of the detector), although some quantities can be computed.

Absorptions (possibly representing leaks) are balanced by multiplicative scattering and spontaneous, including extraneous, sources.

The “open loop” mode of running the experiment is to observe the detector output as a function of any perturbation, normally of the sample.

The “closed loop” mode of running the experiment is to maintain constant the detector output with the control system, for any (not too large) perturbation, and to observe the control parameter instead of the detector output.

The set value of the detector output is a compromise between high for statistical precision and low for the absence of thermal effects.

We want quasi-criticality, see section 3.4. The purpose of quasi-criticality is:
- to obtain a large number of events in a limited time, for statistical precision, even if the spontaneous source is small,
- to submit the sample to collisions with particles coming mostly from the pile, not directly from the spontaneous source.

If the sample has some extent, with quasi-criticality, it will mainly see particle distributions proper to the pile cross sections and almost independent of the spontaneous source.

In a nuclear fission reactor, the particles are neutrons, the pile contains uranium or plutonium, multiplicative scattering is mainly neutron induced fission of uranium 235 or plutonium 239 nuclei and also the (n,2n) scattering on some materials; the (α, n) reaction makes a spontaneous source, an extraneous source can be made of a piece of spontaneously fissile californium 252.

The detector can be a fission or boron ionization chamber (involving ion transport).

6. Experiment model

The collision probability model based on {3, 2, 16} is completed with a geometry: each experimental function, numbered from 2 to 5 in section 5, is localized in one dressed sphere, so that four small dressed spheres occur, as in subsection 4.6. The pile, numbered 1, is represented by a large sphere surrounding all the others, although its precise boundary does not really count (but its volume counts). The quasi-critical limit $\overline{k} \to 1^-$ must be allowed.

In the most symmetric position, the regular tetrahedron (see subsection 1.3), the anisotropy (see subsection 4.6) vanishes .

6.1. Configure

Reference material and geometry:

In[75]:=
```
materialGeometry1 = {{"Pile", .04, 1.01850, 0, 0, 0, 0, 1, □},
  {"Sample", 3, .5, 0, 0, 0, Sqrt[2/3] - 1/(2 Sqrt[6]), .12, .3},
  {"Control", 3, .5, 0, 1/Sqrt[3], 0, -1/(2 Sqrt[6]), .12, .3},
  {"Detector", 3, .5, 0, -1/(2 Sqrt[3]), 1/2, -1/(2 Sqrt[6]), .12, .3},
  {"ExtraSource", 3, .5, 1, -1/(2 Sqrt[3]), -1/2, -1/(2 Sqrt[6]), .12, .3}};
```

The pile has no dressing, so that r_5 does not exist (the placeholder □). x_1, y_1, z_1 do not count and are only used in graphics as a visual reference. Since dressing is a model option, not a physical feature, the Σ, k, q of dressing should be those of the pile, hence not independent, although technically they can be perturbed. Without dressing, last column is not used and can be dropped.

In[•]:=
```
TableForm[materialGeometry1,
  TableHeadings → {Range@5, {"region", Σi, ki, qi, xi, yi, zi, ri, r4+i}}]
```

Out[•]//TableForm=

	region	Σ_i	k_i	q_i	x_i	y_i	z_i	r_i	r_{4+i}
1	Pile	0.04	1.0185	0	0	0	0	1	□
2	Sample	3	0.5	0	0	0	$\sqrt{\frac{2}{3}} - \frac{1}{2\sqrt{6}}$	0.12	0.3
3	Control	3	0.5	0	$\frac{1}{\sqrt{3}}$	0	$-\frac{1}{2\sqrt{6}}$	0.12	0.3
4	Detector	3	0.5	0	$-\frac{1}{2\sqrt{3}}$	$\frac{1}{2}$	$-\frac{1}{2\sqrt{6}}$	0.12	0.3
5	ExtraSource	3	0.5	1	$-\frac{1}{2\sqrt{3}}$	$-\frac{1}{2}$	$-\frac{1}{2\sqrt{6}}$	0.12	0.3

```
configureExperiment ::missingValues =
  "Missing values: one column missing in material and geometry matrix,
    probably the last with dressing radii.";
Clear@configureExperiment;
configureExperiment[OptionsPattern[{
    toPerturb → {},
    materialGeometry → materialGeometry1,
    dressing → True}]] := Catch@(
   Options@buildExperiment = {
      (*tunnel:*)
      toPerturb → OptionValue@toPerturb,
      With[{
        dressing1 = OptionValue@dressing,
        materialGeometry1 = OptionValue@materialGeometry},
       If[dressing1 && Last@Dimensions@materialGeometry1 < 9,
        Message[configureExperiment::missingValues]];
       Sequence @@ {
         dressing → dressing1,
         (*same option name with different format:*)
         materialGeometry →
          (*make list of rules from matrix:*)
          ({names → First@#, Sequence @@ Flatten@{
                   (*Rest/@*)MapThread[threadRule[#1][#2]&, {{x, y, z}, Take[#, {5, 7}]}],
                   threadRule[r]@Join[
                     Extract[#, 8], If[dressing1, Extract[#, 9, Rest], {}]],
                   MapThread[threadRule[#1][#2]&,
                    {{Σ, k, q},
                     If[dressing1,
                         Join[#, Table[First@#, Length@# - 1]], #]&/@Take[#, {2, 4}]}]
                 }}&@Transpose@materialGeometry1)}],
      graphics → True,
      (*this purely graphical effect:*)checkShadow → False,
      beyondGraphics → (*use False to debug graphics:*)True,
      probabilityModel → (*supports thick regions:*)probabilityModel4,
      checkNonNegative (*diagonal probabilities*) → True,
      (*model*)internals (*that you can't directly observe*) → True,
      method → LinearSolve,
      simplifyTime → 10
     };)
```

In[ ]:= `configureExperiment[toPerturb → {k2, Σ3}]`

6.2. Build

In[ ]:=
```
Options @ buildExperiment
```

Out[ ]=
```
{toPerturb → {k2, Σ3}, dressing → True,
 materialGeometry → {names → {Pile, Sample, Control, Detector, ExtraSource}, x1 → 0,
   x2 → 0, x3 → 1/Sqrt[3], x4 → -1/(2 Sqrt[3]), x5 → -1/(2 Sqrt[3]), y1 → 0, y2 → 0, y3 → 0, y4 → 1/2, y5 → -1/2,
   z1 → 0, z2 → Sqrt[2/3] - 1/(2 Sqrt[6]), z3 → -1/(2 Sqrt[6]), z4 → -1/(2 Sqrt[6]), z5 → -1/(2 Sqrt[6]), r1 → 1, r2 → 0.12,
   r3 → 0.12, r4 → 0.12, r5 → 0.12, r6 → 0.3, r7 → 0.3, r8 → 0.3, r9 → 0.3, Σ1 → 0.04,
   Σ2 → 3, Σ3 → 3, Σ4 → 3, Σ5 → 3, Σ6 → 0.04, Σ7 → 0.04, Σ8 → 0.04, Σ9 → 0.04, k1 → 1.0185,
   k2 → 0.5, k3 → 0.5, k4 → 0.5, k5 → 0.5, k6 → 1.0185, k7 → 1.0185, k8 → 1.0185,
   k9 → 1.0185, q1 → 0, q2 → 0, q3 → 0, q4 → 0, q5 → 1, q6 → 0, q7 → 0, q8 → 0, q9 → 0},
 graphics → True, checkShadow → False, beyondGraphics → True,
 probabilityModel →
  Pij → (expM[Sum_{k=2}^{-1+n} Σm[k] (-x[-1+k] + x[k]) + Σi x[1]] Vj xExpM1[Σj (dij - x[-1+n])]) / (4 π dij^2 (dij - x[-1+n])),
 checkNonNegative → True, internals → True, method → LinearSolve}
```

Prepare everything (even the graphics), leaving only the perturbed parameters as symbols to be replaced at run time. Perturbation theory: the fewer elements change, the faster the computation.

In[79]:=
```
buildExperiment ::toPerturbNotList =
  "toPerturb must be List even if it has only one element.";
buildExperiment ::internalsIgnored =
  "You can't get internals if not beyondGraphics.";
buildExperiment ::firstIsPile =
  "Warning: first region considered as pile should be named \"Pile\".";
buildExperiment ::anisotropyNotSymmetric =
  "Warning: anisotropy matrix not symmetric.";
Clear @ buildExperiment;
(*OptionsPattern [] cannot take Options from last step:*)
buildExperiment @ option : (_Rule | _RuleDelayed) ... :=
 (*for debbugging, Throw what you want to see:*)
 Catch @ (
   Options @ runExperiment = Flatten @ Rest @ Reap[
       With[{
         toPerturb1 = toPerturb /. {option} /. Options @ buildExperiment,
         (*material and geometry suboptions:*)
         materialGeometry1 = materialGeometry /. {option} /.
           Options @ buildExperiment
```

```
},
Switch[(*must be list even with only one element*)
 toPerturb1 , Except @ _List ,
 Throw @ Message[buildExperiment ::toPerturbNotList ]];
(*tunnel to next step:*)
Sow @ {toPerturb → toPerturb1 ,
  (*reference values of perturbed parameters are the only material
   and geometry information directly passed to run:*)
  FilterRules [materialGeometry1 , toPerturb1 ]};
With[{
  (*perturbed parameter values must not be used now because they
   may change at run time*)
  options1 = FilterRules [{option}, Except @ Apply[Alternatives , toPerturb1 ]]},
 With[
  {options2 = FilterRules [materialGeometry1 ,
      Except @ Apply[Alternatives , toPerturb1 ]]},
  With[{
    layout1 = MapIndexed [{#1 , Apply[Sequence , {{x#, y#, z#}, #} &@ only @ #2]} &,
        names /. options1 /. options2 ] /. options1 /. options2 },
   (*example layout1 ==
    {{"Pile",{0,0,0},1},{"Sample",{0,0,1},2}…}*)
   With[{
     (*number of spheres except the pile:*)
     n0 = Length @ layout1 - 1,
     detector1 = only @ Cases[layout1 , {"Detector", ___ , n_Integer } :→ n],
     distances = ((*first must be pile*)
       If[layout1 [[1, 1]] =!= "Pile", Message[buildExperiment ::firstIsPile ]];
       Apply[d### &, Sort[Last /@ #1]] → EuclideanDistance @@ First /@ # & /@
        Subsets[Rest /@ Rest @ layout1 , {2}]),
     dressing1 = dressing /. options1 /. Options @ buildExperiment ,
     method1 = method /. options1 /. Options @ buildExperiment ,
     internals1 = internals /. options1 /. Options @ buildExperiment },
    If[graphics /. options1 /. Options @ buildExperiment ,
     Sow @ {
       planeView → (If[internals /. options1 /. Options @ buildExperiment , #,
            # /. GrayLevel _i_ :→ (*all white since fluxes unknown:*)
                GrayLevel @ 1 /. options1 /. options2
           ] &@ Hold[Graphics ][{
              {Style[Disk @ ### , GrayLevel 1], Circle @ ###} &@
               Sequence [Most @ layout1 [[1, 2]], r1],
              Text[StringTake [layout1 [[1, 1]], 1] <> "1", {#, #} &[.9 r1]],
              Arrow[{{#, #} &[.85 r1], {#, #} &[Sqrt @ 2 / 2 r1]}],
              With[{checkShadow1 = checkShadow /. options1 /.
                   Options @ buildExperiment },
```

```
                    If[checkShadow1 ,
                      (*If z2 is perturbed , it is not known now => Hold :*)
                      Hold @ Composition [Rest /@ # &, SortBy [#, 1. First [#] &] &],
                      Identity ][Function [{label , center , i},
                          {
                           (*add z for sorting *)
                           If[checkShadow1 , Last @ center ,
                            Unevaluated @ Sequence []],
                           If[dressing1 , Apply [Sequence ,
                             {Style [Disk @ ##, GrayLevel i+n0],
                                 Style [Circle @ ##, Dashed ]} & @
                              Sequence [Most @ center , r i+n0]],
                            Unevaluated @ Sequence []],
                           Apply [Sequence , {Style [Disk @ ##, GrayLevel i],
                                Circle @ ##} & @ Sequence [Most @ center , r i]],
                           Style [{Text [StringTake [label , 1] <> ToString @ i ,
                               Most @ center ],
                              If[dressing1 , Text [ToString [i + n0],
                                Most @ center + {(r i + r i+4) / 2, 0}],
                                Unevaluated @ Sequence []]}, Background → White ]}] @@
                         # & /@
                      Reverse @ Rest @ layout1 ](*,
                   Text [Grid [Reverse @ Drop [#, {2}] & /@ SortBy [layout1 , Last],
                     Frame → False , Spacings → 1, Alignment → Left],
                    r 1 1.05 {-1, -1}, {-1, -1}]*)} /. options1 /. options2 ])
                }];
              If[Not[beyondGraphics /. options1 /. Options @ buildExperiment ],
               If[internals1 , Message [buildExperiment :: internalsIgnored ]],
               With[{
                  volumes = threadRule [V][(*pile contains others :*)4 / 3 π {r 1^3 - Sum[r i+1^3, {i, 1, n0}],
                        Sequence @@ Table [r i+1^3, {i, n0}], Sequence @@
                         If[dressing1 , Table [r i+1+n0^3 - r i+1^3, {i, n0}], {}]}] /. options1 /.
                    options2 },
                 With[{expandProbability3 =
                    expandProbability1 [probabilityModel /. options1 /.
                          Options @ buildExperiment , n0, dressing → dressing1 ] /.
                        volumes /. distances /. options1 /. options2 },
                  With[{
                    expandProbability4 = Map[# /. (P i_j_ → {mean_ , _}) :→ P i j → mean &,
                      expandProbability3 , {2}],
                    anisotropy1 = Map[# /. (P i_j_ → {_, meanDeviation_ }) :→
```

```
        αi j → meanDeviation  &, expandProbability3  , {2}],
 dim = If[dressing1 , 2, 1] n0 + 1},
With[{expandProbability5  = simplify [simplifyTime  /. options1 /.
       Options @ buildExperiment , "Pij"]@
     MapThread [Sort @{#1 /. #2, Sequence @@ #2} &, {
       (*Simplify @*)solveDiagonal [dim, checkNonNegative  →
         (*to save ByteCount , will check numerically  at run :*)
         False ],
       expandProbability4  }],
  anisotropy2  =
   (*fill  the  diagonal :*)Map[Last , #, {2}] &[
    MapThread [Sort @* Append , {#, threadRule [α][
           (*pessimistic  estimate :*)(*Total *)
           0 & /@ Map[Last , #, {2}]] /. αi_ → αi i}] & @ anisotropy1 ]
 },
 If[Not @ And @@ Map[# == 0 &, Flatten @
     Chop[anisotropy2  - Transpose @ anisotropy2 ]],
  Message @ buildExperiment ::anisotropyNotSymmetric  ];
 If[internals1 ,
  Sow @ {
    (*tunnel :*)
    detector → detector1 ,
    dressing → dressing1 ,
    (*:tunnel.. *)
    stochasticMatrix  → Map[Last , expandProbability5  , {2}],
    reciprocitySUTMatrix  →
     (Table[ReleaseHold @ reciprocity , {i, dim - 1}, {j, i + 1, dim}] /.
           Flatten @ expandProbability5  /. volumes /. options1 /.
       options2 ),
    anisotropyMatrix  → (anisotropy2  /. volumes /. options1 /.
       options2 )}
 ];
 If[checkNonNegative  /. options1 /. Options @ buildExperiment ,
  (*can be used even if not internals *)
  Sow[stochasticMatrixDiagonal  → Last /@
     Diagonal @ expandProbability5  ]];

 With[{matrixVector1  = Transpose @ (*replace  by line  is  enough *)
     MapThread [ReplaceAll , {
       Transpose [matrixVector [dim, method → method1 ] /. options1 /.
         options2 ],
       If[method1 === Inverse ,
          (*replace  only by number  or else  ByteCount
            is  too  large :*)Select [#, NumberQ @* Last] & /@ #,
```

```
                #
               ] & @ Transpose @ expandProbability5 }]},
          With[{solutionVector = Switch[method1, LinearSolve,
             Hold[Apply][
              LinearSolve, matrixVector1 ],
             Inverse,
             (*don't replace the Pij yet to save ByteCount
              if no internals *)
             Map[(*hangs *)(*Simplify *)Identity, Print @ "Solving ";
              Inverse @ First @ matrixVector1 .Last @ matrixVector1 ],
             FixedPoint,
             Hold[FixedPoint][
              Function[Plus[matrix1 .#, vector1 ]] /.
               {matrix1 → First @ matrixVector1, vector1 →
                 Last @ matrixVector1 }, Table[0, dim]]
            ]},
           With[{
             replaceProbabilities = Function[Composition[
                 (Print @ "Replacing "; ReplaceAll @ ##) &][
                #, Flatten @ expandProbability5 ]]},
            If[Not @ internals1,
             Sow @ {observable → If[
                 method1 === Inverse,
                 (*hangs if perturb x*)simplify[70, "after replacing "] @
                  replaceProbabilities @
                   (*hangs, try factoring polynomials ?*)
                   solutionVector [[detector1 ]],
                 Hold[Part][solutionVector, detector1 ]
                ]},
             Sow @ {
               solution → If[method1 === Inverse, threadRule [R] @
                  (*hangs if perturb x*)simplify[500,
                    "after replacing "] @ replaceProbabilities @
                    solutionVector, Hold[Thread][
                   Table[Rᵢ, {i, dim}] → solutionVector ]],
               (*save ByteCount or Timing => use solutionVector
                    only once =>
                    use solution instead =>
                    run will need to apply rules one more time*)
               observable → Hold[ReplaceAll][R_detector1, solution ],
               fluxes → Hold[ReplaceAll][Table[Rᵢ / (Vᵢ Σᵢ), {i, dim}] /.
                     volumes /. options1 /. options2, solution ],
               amplificationMultiplication → Hold[ReplaceAll][
                 expandSum[{|R| / |q|, k̄}, dim] /. options1 /. options2,
```

```
solution ]}]]]]]]]]]]]]]);
```

```
In[ ]:= configureExperiment [toPerturb → {k2, Σ3}]
```

Default options can be overridden, for flexibilty:

```
In[ ]:= buildExperiment [toPerturb → {k2, k3}]
```

```
In[ ]:= toPerturb /. Options @ runExperiment
```

```
Out[ ]= {k2, k3}
```

or inherited, for brevity:

```
In[ ]:= buildExperiment []
        toPerturb /. Options @ runExperiment
```

```
Out[ ]= {k2, Σ3}
```

6.3. Run

```
In[320]:= configureExperiment [toPerturb → {k2, Σ3}]
```

```
In[321]:= buildExperiment [simplifyTime → 0]
```

```
In[322]:= First /@ Options @ runExperiment
```

```
Out[322]= {toPerturb , Σ3, k2, planeView , detector , dressing , stochasticMatrix ,
  reciprocitySUTMatrix , anisotropyMatrix , stochasticMatrixDiagonal ,
  solution , observable , fluxes , amplificationMultiplication }
```

In[323]:=
```
Options @ runExperiment // Short[#, 30] &
```

Out[323]//Short=
```
{toPerturb → {k2, Σ3}, Σ3 → 3, k2 → 0.5,
 planeView → Hold[Graphics][{{Style[Disk[{0, 0}, 1], GrayLevel 1], Circle[{0, 0}, 1]},
    Text[P1, {0.9, 0.9}], Arrow[{{0.85, 0.85}, {1/Sqrt[2], 1/Sqrt[2]}}],
    {{Style[Disk[{-1/(2 Sqrt[3]), -1/2}, 0.3], GrayLevel 9], Circle[{-1/(2 Sqrt[3]), -1/2}, 0.3],
      Style[Disk[{-1/(2 Sqrt[3]), -1/2}, 0.12], GrayLevel 5], Circle[{-1/(2 Sqrt[3]), -1/2}, 0.12],
      {Text[E5, {-1/(2 Sqrt[3]), -1/2}], Text[9, {-0.0786751, -1/2}]}},
     {Style[Disk[{-1/(2 Sqrt[3]), 1/2}, 0.3], GrayLevel 8], Circle[{-1/(2 Sqrt[3]), 1/2}, 0.3],
      Style[Disk[{-1/(2 Sqrt[3]), 1/2}, 0.12], GrayLevel 4], Circle[{-1/(2 Sqrt[3]), 1/2}, 0.12],
      {Text[D4, {-1/(2 Sqrt[3]), 1/2}], Text[8, {-0.0786751, 1/2}]}},
     {Style[Disk[{1/Sqrt[3], 0}, 0.3], GrayLevel 7], Circle[{1/Sqrt[3], 0}, 0.3],
      Style[Disk[{1/Sqrt[3], 0}, 0.12], GrayLevel 3], Circle[{1/Sqrt[3], 0}, 0.12],
      {Text[C3, {1/Sqrt[3], 0}], Text[7, {0.78735, 0}]}}, {Style[Disk[{0, 0}, 0.3], GrayLevel 6],
      Circle[{0, 0}, 0.3], Style[Disk[{0, 0}, 0.12], GrayLevel 2],
      Circle[{0, 0}, 0.12], {Text[S2, {0, 0}], Text[6, {0.21, 0}]}}}],
 ≪7≫, observable → Hold[ReplaceAll][R4, solution],
 fluxes →
  Hold[ReplaceAll][
   {6.00985 R1, 46.0518 R2, 138.155 R3/Σ3, 46.0518 R4, 46.0518 R5,
    236.163 R6, 236.163 R7, 236.163 R8, 236.163 R9}, solution],
 amplificationMultiplication → Hold[ReplaceAll][{R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9,
    (1.0185 R1 + k2 R2 + 0.5 R3 + 0.5 R4 + 0.5 R5 + 1.0185 R6 + 1.0185 R7 + 1.0185 R8 + 1.0185 R9)/
     (R1 + R2 + R3 + R4 + R5 + R6 + R7 + R8 + R9)}, solution]}
```

In[316]:=
```
solution /. Options @ runExperiment // Short[#, 12] &
```

Out[316]//Short=
```
Hold[Thread][{R1, R2, R3, R4, R5, R6, R7, R8, R9} → Hold[Apply][LinearSolve ,
    {{{-0.0169847 , 0.0181974 , 0.0181974 , 0.0181974 , 0.0181974 , 0.158509 , 0.158509 ,
        0.158509 , 0.158509 }, ≪7≫, {0.00403372 , 0.000183362 , 0.000183362 ,
        0.000183362 , 0.00265583 , 0.000996799 , 0.000996799 , 0.000996799 , -0.176489 }},
     {-0.0363948 , -0.000815923 , -0.000815923 , ≪4≫, -0.000366724 , -0.00531166 }}]]
```

Replace the perturbed parameters by numerical values, do some tests (optional but safer), and solve (or apply the general solution already obtained), as fast as possible:

```
runExperiment ::negativeProbability = "probabilities can't be normalized ";
runExperiment ::negativeFlux = "supercritical negative flux ";
Clear @ runExperiment ;
runExperiment @
  option :
   (_ → _?(*non numeric see FindRoot documentation Examples possible issues :*)
        NumericQ ) ... :=
 Catch[(Options @ reportExperiment = DeleteCases [Flatten @ #2 , x_ → x_];
     #1) & @@ Reap[
    (*tunnel :*)
    Sow @ {
      detector → (detector /. Options @ runExperiment ),
      dressing → (dressing /. Options @ runExperiment )
     };
    (*:tunnel *)
    With[{options1 = (*perturbed parameters :*)
        FilterRules [{option}, Apply[Alternatives ,
          toPerturb /. Options @ runExperiment ]]},
     With[{stochasticMatrixDiagonal1 =
         stochasticMatrixDiagonal /. options1 /. Options @ runExperiment /. options1 /.
          Options @ runExperiment },
       Switch[(*not list if build[checkNonNegative →False]:*)
        stochasticMatrixDiagonal1 , _List ,
        If[hasNegativeElement @ stochasticMatrixDiagonal1 ,
         Message @ runExperiment ::negativeProbability ]]];
      Sow @ {
        (*repeated rules easier for debug :*)
        stochasticMatrix →
         (stochasticMatrix /. options1 /. Options @ runExperiment /. options1 /.
           Options @ runExperiment ),
        reciprocitySUTMatrix →
         (reciprocitySUTMatrix /. options1 /. Options @ runExperiment /. options1 /.
           Options @ runExperiment ),
        anisotropyMatrix →
         (anisotropyMatrix /. options1 /. Options @ runExperiment /. options1 /.
```

```
      Options @ runExperiment )};
    With[
     {solution1 = ReleaseHold [
        solution /. options1 /. Options @ runExperiment  /. options1 /.
         Options @ runExperiment ]},
     Sow @ {
       (*solution  vector *)
       collisionRates  → Map[Last , solution1 ],
       amplificationMultiplication   →
        ReleaseHold [
         amplificationMultiplication   /. options1 /. Options @ runExperiment  /.
            solution → solution1 /. options1 /. Options @ runExperiment ]};
     With[
      {fluxes1 = ReleaseHold [
          fluxes /. options1 /. Options @ runExperiment  /. solution → solution1 /.
            options1 /. Options @ runExperiment ]},
       Sow @ {
          fluxes → fluxes1 ,
          planeView → ReplaceAll [
            planeView /. options1 /. Options @ runExperiment  /.
              threadRule [GrayLevel ][
               Map[GrayLevel , If[Quiet[hasNegativeElement  @ fluxes1 ,
                    {Select :: normal }],
                  Message @ runExperiment :: negativeFlux ;
                  (*all  white :*)
                  1 & /@ # &,
                  rescaleLog [#, grayLevelInterval ] &
                 ]@ fluxes1 ]
              ] /. options1 /. Options @ runExperiment , Hold → Identity ]}];
     ReleaseHold [
      observable  /. options1 /. Options @ runExperiment  /. solution → solution1 /.
        options1 /. Options @ runExperiment ]
    ]]]]
```

Just as in reality, the observable is raw number (all the rest is interpretation):

In[324]:=
```
runExperiment []
```

Out[324]=
```
0.660061
```

In[325]:=
```
toPerturb /. Options @ runExperiment
```

Out[325]=
```
{k2, Σ3}
```

In[326]:=
```
Thread [(toPerturb /. Options @ runExperiment ) → {1, 3}]
```

Out[326]=
```
{k2 → 1, Σ3 → 3}
```

In[329]:= `runExperiment [k2 → 1, Σ3 → 3]`

runExperiment : supercritical negative flux

Out[329]= -0.235428

In[330]:= `runExperiment [k2 → .9, Σ3 → 3]`

Out[330]= 33.0008

6.4. Feedback

In[331]:= `configureExperiment [toPerturb → {x2, Σ3, z2}]`

In[332]:= `buildExperiment [internals → True, simplifyTime → 0]`

In[333]:= `First /@ Options @ runExperiment`

Out[333]= {toPerturb , x2, z2, Σ3, planeView , detector , dressing , stochasticMatrix ,
reciprocitySUTMatrix , anisotropyMatrix , stochasticMatrixDiagonal ,
solution , observable , fluxes , amplificationMultiplication }

In[334]:= `Σ3 /. Options @ runExperiment`

Out[334]= 3

```
(*runExperiment [option___ ,feedback →None |feedback :→None]:=runExperiment @option ;
*)
runExperiment ::wrongControl =
  "Only a perturbed parameter can be used for feedback .";
runExperiment ::cannotControl =
  "Feedback requires at least two perturbed parameters .";
(*findRoot can be FindRoot or bisection for example :*)
runExperiment [option___ ,
  feedback :→ findRoot_ [{feedbackParameter_ , init___ }, option1___ ]] :=
 With[{toPerturb1 = toPerturb /. Options @ runExperiment },
  Which[Not @ MatchQ [feedbackParameter , Apply[Alternatives , toPerturb1 ]],
   Message @ runExperiment ::wrongControl ,
   Length @ toPerturb1 < 2,
   Message @ runExperiment ::cannotControl ];
  With[{
    option2 = Sequence @@ FilterRules [{option },
       Alternatives @@ DeleteCases [toPerturb1 , feedbackParameter ]],
    R0 = runExperiment [],
    option3 = AccuracyGoal → $MachinePrecision / 2
   },
   With[
    {root = only @ findRoot [runExperiment [option2 , feedbackParameter → root ] - R0 ,
         {root , init}, option1 , option3 ] /. root → feedbackParameter },
    (*possible optional improvement of next
     guess : assuming that parameters vary continuously as with Manipulate ,
    update Options @run with last used parameter values *)
    Options @ reportExperiment =
     Join[FilterRules [Options @ reportExperiment ,
       Except [feedbackRoot | feedbackSuccess ]], {feedbackRoot → root ,
       feedbackSuccess → Abs[runExperiment [option2 , root ] - R0 ] <
         10 ^ (- AccuracyGoal /. {option1 } /. {option3 })}];
    Last @ root
   ]]]
```

In[335]:=
```
runExperiment [x2 → .5 ,
 feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5 , 5}, AccuracyGoal → 13]]
```

Out[335]=
```
3.31386
```

In[336]:=
```
{feedbackRoot , feedbackSuccess } /. Options @ reportExperiment
```

Out[336]=
```
{Σ3 → 3.31386 , True }
```

Impossible accuracy goal:

In[339]:=
```
runExperiment [x2 → .5 ,
 feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5 , 5}, AccuracyGoal → 20]]
```

FindRoot : The line search decreased the step size to within tolerance specified by AccuracyGoal and PrecisionGoal but was unable to find a sufficient decrease in the merit function . You may need more than MachinePrecision digits of working precision to meet these tolerances .

Out[339]= 3.31386

In[338]:=
```
{feedbackRoot , feedbackSuccess } /. Options @ reportExperiment
```

Out[338]= {Σ3 → 3.31386 , False }

Impossible problem:

In[340]:=
```
runExperiment [x2 → .7 , feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5 , 5}]]
```

runExperiment : supercritical negative flux

runExperiment : supercritical negative flux

runExperiment : supercritical negative flux

General : Further output of runExperiment ::negativeFlux will be suppressed during this calculation .

FindRoot : The point {0.5 } is at the edge of the search region {0.5 , 5.} in coordinate 1 and the computed search direction points outside the region .

Out[340]= 0.5

In[341]:=
```
{feedbackRoot , feedbackSuccess } /. Options @ reportExperiment
```

Out[341]= {Σ3 → 0.5 , False }

The feedback parameter (by default the last perturbed parameter) can be changed to any perturbed parameter:

```
runExperiment [x2 → .25 ,
 feedback :→
  FindRoot [{(*mistake , not perturbed :*)z3, z3 /. Options @ runExperiment , -.5 , 5}]]
```

runExperiment : Only a perturbed parameter can be used for feedback .

FindRoot : Value z3 in search specification {root , z3 /. Options [runExperiment], −0.5 , 5} is not a number or array of numbers .

FindRoot : The variable z3 cannot be localized so that it can be assigned to numerical values .

Out[342]=
```
FindRoot [runExperiment [Sequence [x2 → 0.25], z3 → z3] - 0.660061 ,
 {z3, z3 /. Options [runExperiment ], -0.5 , 5}, AccuracyGoal → 7.97729 ]
```

In[344]:=
```
runExperiment [x2 → .25 , feedback :→ FindRoot [{z2, z2 /. Options @ runExperiment , -.5 , 5}]]
```

Out[344]= 0.610637

Timing

In[347]:=
```
runExperiment [x2 → .5, feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5, 5}]] //
 Timing
```

Out[347]=
```
{0.976578 , 3.31386 }
```

Slower with internals:

In[348]:=
```
buildExperiment [internals → False, simplifyTime → 0]
```

In[349]:=
```
runExperiment [x2 → .5, feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5, 5}]] //
 Timing
```

Out[349]=
```
{0.354079 , 3.31386 }
```

To speed up root finding without losing internals for report, cheat Options@runExperiment (with dynamical scoping or block):

In[354]:=
```
buildExperiment [internals → True, simplifyTime → 0]
```

In[355]:=
```
With[{savedOptions = Options @ runExperiment },
 Options @ runExperiment =
  FilterRules [savedOptions , {toPerturb , Apply[Sequence , toPerturb /. savedOptions ],
    (*no test faster :*)(*stochasticMatrixDiagonal ,*)solution , observable }];
 Print[
  runExperiment [x2 → .5, feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5, 5}]] //
   Timing ];
 Options @ runExperiment = savedOptions ;
]
```

```
{0.216544 , 3.31386 }
```

Same outside of block:

In[356]:=
```
runExperiment [x2 → .5, feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5, 5}]] //
 Timing
```

Out[356]=
```
{0.991131 , 3.31386 }
```

6.5. Report

```
configureExperiment [toPerturb → {x2, Σ3, z2}]
buildExperiment [internals → True, simplifyTime → 0]
```

Purposely bad case:

```
runExperiment [x2 → .7, feedback :→ FindRoot [{Σ3, Σ3 /. Options @ runExperiment , .5, 5}]]
{feedbackRoot , feedbackSuccess } /. Options @ reportExperiment
```

```
runExperiment : supercritical negative flux
runExperiment : supercritical negative flux
runExperiment : supercritical negative flux
General : Further output of runExperiment ::negativeFlux will be suppressed during this calculation .
FindRoot : The point {0.5} is at the edge of the search region {0.5, 5.} in coordinate 1 and the computed search direction points outside the region .
```

Out[359]= 0.5

Out[360]= {Σ3 → 0.5, False}

In[361]:=
```
First /@ Options @ reportExperiment
```

Out[361]=
```
{detector , dressing , stochasticMatrix , reciprocitySUTMatrix ,
 anisotropyMatrix , collisionRates , amplificationMultiplication ,
 fluxes , planeView , feedbackRoot , feedbackSuccess }
```

Generate a nice report:

In[280]:=
```
reportExperiment ::shortIgnored = "Without dressing , all internals are shown ."
Clear @ reportExperiment ;
reportExperiment @ OptionsPattern [
    short → False] := Catch @ ReleaseHold [Hold[
      (*global layout :*)
      Column [{(*PijRΦ *)#3 ,
              Row[{Column [{
                   (*R/q k*)#2 ,
                   (*anisotropy *)#5 ,
                   (*BarChart *)#4}, Alignment → Center , Spacings → 0], Spacer @ 0,
                (*planeView *)#1 , Spacer @ 0
               }]} & @@ # , Alignment → Left , Spacings → 0] &@
       (*flat list of items to return :*)
       {
        Graphics @
         Append [Replace [planeView , _Symbol → Graphics @{Text @ "missing plane view"}], {
           ImageSize → {#, #} &@ 230 , Axes → True , AxesLabel → {"x", "y"},
           Ticks → None ,
           ImagePadding → {{10, 20}, {10, 20}}, PlotRange → 1.03 {{-1, 1}, {-1, 1}}}],

        Apply [Sequence , Switch [amplificationMultiplication , _Symbol ,
             Table ["missing value ", {Length [Hold /@ Unevaluated @ #]}], _, #]] &@
         Unevaluated [
          {
           Style [#, FontSize → 13] &@
```

```
 Grid[{{"|R|/|q|", "k̄"}, {constantLength [#1 , 6], constantLength [#2 , 5]} & @@
    amplificationMultiplication  }, Frame → All , Spacings → {Automatic , .3}],

With[{dim = Length @ First @ #},
   With[{
     n0 = (dim - 1) / If[dressing , 2 , 1],
     short1 = OptionValue @ short && If[Not @ dressing ,
        Message @ reportExperiment ::shortIgnored ;
        False , True]
    }, Composition [
      Pane[#, {Automatic , 130}, Alignment → Bottom] &,
      Style[#, FontSize → If[OptionValue @ short , 13 , 11]] &,
      Grid[#, (*gray background  for the observable ,
        where reciprocity  fails*)Background →
         {None , None , Append[Rule[#, GrayLevel @ .6] & /@ Join[
              (*diagonal  for lisibility :*)
              Table[{i, i} + 1, {i, dim}],
              (*where reciprocity  fails :*)Apply[Sequence ,
               {#, Reverse /@ #} &[Cases[#, (x_ → False) :> x + {1, 1}] &@
                       Flatten[#, 1] &@ MapIndexed [
                     {0, Last @ #} + First @ # &@ #2 → #1 &, #, {2}] &@
                  If[short1 , Drop[#, -n0, -n0] &, Identity ][
                   reciprocitySUTMatrix  /. Options @ reportExperiment ]]]],
          (*observable :*)
          {If[short1 || Not @ dressing , 1, 2] n0 + 3, detector + 1} → Which[
            feedbackSuccess  === True , Lighter @ Green ,
            Head @ feedbackRoot  === Rule , Lighter @ Red ,
            True , Lighter @ Orange ]]},
        Dividers → {{None , {True}, None}, {2 → True , -2 → False ,
           -3 → True}},
        Frame → False ,
        ItemSize → {Full , {Full , 1.2 , {Full}, 1.2 , 1.9}},
        Spacings → {.2 , {0, 0, {0}, 0, -.8 , 0}}] &,
      (*number  rows :*)
      Composition [Transpose ,
       Prepend[#, {"P_{i\\j}", Sequence @@ Range[Length @ First @ # - 3],
          "R_j", "Φ_j"}] &, Transpose ],
      (*number  columns :*)
      Prepend[#, Range[Length @ First @ #]] &,
      (*add collision  rates :*)
      Append[#, constantLength [#, 5] & /@ Drop[fluxes ,
          If[short1 , -n0, None]]] &,
      Append[#, constantLength [#, 4] & /@ Drop[collisionRates ,
          If[short1 , -n0, None]]] &,
```

```
            Map[constantLengthProbability  , #, {2}] &,
            If[short1 , Drop[#, -n0, -n0] &, Identity ]
           ]@#]] &@ stochasticMatrix  ,

        Panel[#, Style["dB[Φj/Φ1]", FontSize → 11], Bottom , ContentPadding → False ,
          Background → White ] &@
         Pane[If[And @@ Positive /@ fluxes , dBBarChart [fluxes ,
            AxesLabel → {"j", None}, AxesOrigin → {Automatic , 0},
            PlotRange → {Automatic , (*Automatic *){-12 , 8}}, Background → White ]],
          {200 , 100}],

        Composition [Pane[#, {Automatic , 30}, Alignment → Bottom ] &,
          Style[#, FontSize → 11] &]@
         Grid[List @{"anisotropy ", constantLength [#, 3] &@ Max @
              Rest @ First[anisotropyMatrix ]}, Frame → All]
       }]
    }](*/.{option }*) /. Options @ reportExperiment ]
```

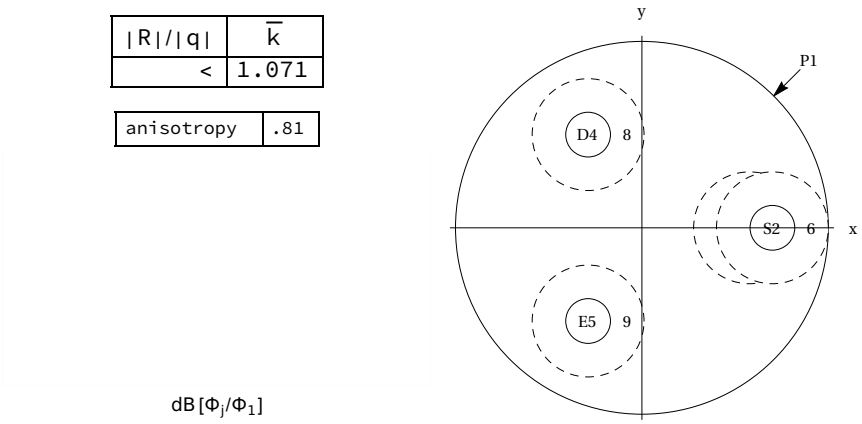
In[362]:= `reportExperiment []`

Out[362]=

$P_{i\setminus j}$	1	2	3	4	5	6	7	8	9
1	97-2	40-4	12-4	40-4	40-4	37-4	53-4	31-4	31-4
2	31-3	96-2	27-5	42-5	42-5	53-4	61-5	17-5	17-5
3	57-3	16-4	93-2	11-4	11-4	82-5	72-4	50-5	50-5
4	31-3	42-5	18-5	96-2	82-5	17-5	37-5	53-4	37-5
5	31-3	42-5	18-5	82-5	96-2	17-5	37-5	37-5	53-4
6	14-2	27-3	70-5	85-5	85-5	82-2	20-4	35-5	35-5
7	21-2	31-4	61-4	19-4	19-4	20-4	77-2	98-5	98-5
8	12-2	85-5	42-5	27-3	19-4	35-5	98-5	84-2	98-5
9	12-2	85-5	42-5	19-4	27-3	35-5	98-5	98-5	84-2
R_j	<	<	<	<	1.72	<	<	<	<

|R|/|q|	$\overline{k}$
<	1.071

anisotropy	.81



Without graphics or internals (for intensively looping functions like Plot, FindRoot):

In[363]:= `buildExperiment [toPerturb → {Σ2}, graphics → False , internals → False ]`

simplify : Simplifying Pij 10 seconds

In[364]:=
```
First /@ Options @ runExperiment
```

Out[364]=
```
{toPerturb , Σ2, stochasticMatrixDiagonal  , observable }
```

Run is fast:

In[366]:=
```
runExperiment [] // Timing
```

Out[366]=
```
{0.000911 , 0.660061 }
```

In[367]:=
```
buildExperiment [toPerturb → {x2, y2}, graphics → True , internals → True]
```

simplify : Simplifying Pij 10 seconds

simplify : Simplify failed time constraint

The starting point is stationary by symmetry, hence the initial divergence and only one solution is found:

In[368]:=
```
runExperiment [x2 → .1, feedback :→ FindRoot [{y2, y2 /. Options @ runExperiment , -.5, +.5}]]
```

Out[368]=
```
-0.0716708
```

In[369]:=
```
runExperiment [x2 → 0.1 , y2 → -0.07167084653771753`  ]
```

Out[369]=
```
0.660061
```

The symmetric solution:

In[371]:=
```
runExperiment [x2 → 0.1 , y2 → 0.07167084653771753`  ]
```

Out[371]=
```
0.660061
```

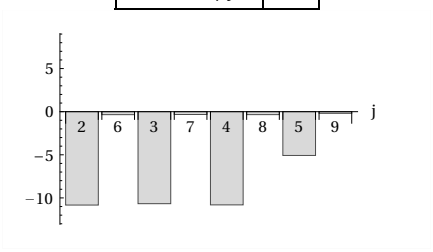
In[372]:=
```
reportExperiment []
```

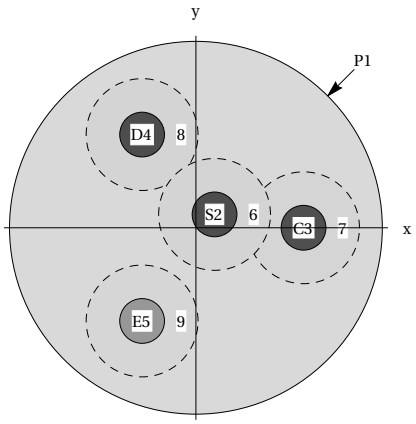
Pi\j	1	2	3	4	5	6	7	8	9
1	97-2	47-4	49-4	47-4	45-4	40-4	42-4	40-4	37-4
2	36-3	95-2	91-5	81-5	71-5	53-4	42-5	37-5	31-5
3	38-3	91-5	95-2	82-5	82-5	42-5	53-4	37-5	37-5
4	36-3	81-5	82-5	95-2	82-5	37-5	37-5	53-4	37-5
5	35-3	71-5	82-5	82-5	96-2	31-5	37-5	37-5	53-4
6	16-2	27-3	22-4	19-4	16-4	81-2	12-4	98-5	78-5
7	17-2	22-4	27-3	19-4	19-4	12-4	80-2	98-5	98-5
8	16-2	19-4	19-4	27-3	19-4	98-5	98-5	81-2	98-5
9	15-2	16-4	19-4	19-4	27-3	78-5	98-5	98-5	82-2
Rj	60.9	.657	.682	.66	2.47	1.43	1.44	1.43	1.48

Out[372]=

|R|/|q|	$\bar{k}$
71.16	.9859

anisotropy	.14





In[373]:=
```
buildExperiment [toPerturb → {x2, y2}, graphics → True , internals → False ]
```

simplify : Simplifying Pij 10 seconds

simplify : Simplify failed time constraint

In[374]:=
```
runExperiment [x2 → .1 , feedback :→ FindRoot [{y2, y2 /. Options @ runExperiment , -.5 , +.5}]]
```

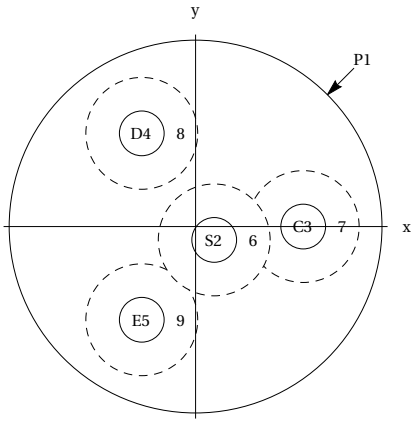
Out[374]=
```
-0.0716708
```

Without internal, all that can be checked is the sample position:

In[375]:=
```
reportExperiment []
```

```
missing value
```

Out[375]=
```
missing value
missing value
missing value
```



The Inverse methods works well with multiplication factor or spontaneous source (as opposed to cross section) perturbations, because the stochastic matrix depends simply on them:

In[376]:=
```
buildExperiment [toPerturb → {q2, q5}, graphics → False , internals → False ,
 method → Inverse ]
```

simplify : Simplifying Pij 10 seconds

Solving

Replacing

simplify : Simplifying after replacing 70 seconds

and run is faster:

In[379]:=
```
runExperiment [] // Timing
```

Out[379]=
```
{0.000323 , 0.660061 }
```

In[380]:=
```
buildExperiment [toPerturb → {k3, q5}, graphics → False , internals → False ,
 method → Inverse ]
```

simplify : Simplifying Pij 10 seconds

Solving

Replacing

simplify : Simplifying after replacing 70 seconds

In[382]:=
```
runExperiment [] // Timing
```

Out[382]=
```
{0.000354 , 0.660061 }
```

7. Consistency checks

As the experiment is simulated, it is its own model. Comparing the experiment and the model yields total agreement by hypothesis. All that can be checked is the model *self-consistency*.

7.1. Reciprocity

Thick sample

In[383]:=
```
configureExperiment [dressing → True ,
 materialGeometry → {{"Pile", .04, 1.01850, 0, 0, 0, 0, 1, □},
  {"Sample", 9, .5, 0, 0, 0, Sqrt[2/3] - 1/(2 Sqrt[6]), .12, .22},
  {"Control", 3, .5, 0, 1/Sqrt[3], 0, -1/(2 Sqrt[6]), .12, .22},
  {"Detector", 3, .5, 0, -1/(2 Sqrt[3]), 1/2, -1/(2 Sqrt[6]), .12, .22},
  {"ExtraSource", 3, .5, 1, -1/(2 Sqrt[3]), -1/2, -1/(2 Sqrt[6]), .12, .22}}]
```

In[384]:=
```
buildExperiment [graphics → False]
```

```
... simplify : Simplifying  Pij  10 seconds
```

In[385]:=
```
runExperiment []
```

Out[385]=
```
0.227336
```

In[386]:=
```
reciprocitySUTMatrix  /. Options @ reportExperiment
```

Out[386]=
```
{{False , True , True , True , True , True , True , True},
 {False , False , False , False , False , False , False},
 {True , True , True , True , True , True}, {True , True , True , True , True},
 {True , True , True , True}, {True , True , True}, {True , True}, {True}}
```

The collision probabilities that failed the reciprocity test have gray background (the diagonal is always gray for reading support but reciprocity can't fail there):

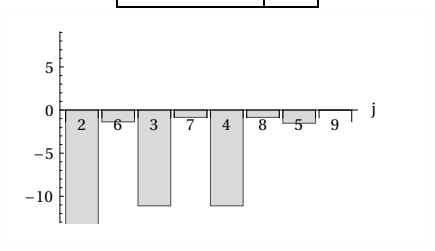
In[387]:= **reportExperiment [short → False]**

$P_{i\setminus j}$	1	2	3	4	5	6	7	8	9
1	97-2	70-4	47-4	47-4	47-4	11-4	11-4	11-4	11-4
2	18-3	98-2	40-5	40-5	40-5	14-4	57-6	57-6	57-6
3	36-3	12-4	96-2	82-5	82-5	12-5	29-4	12-5	12-5
4	36-3	12-4	82-5	96-2	82-5	12-5	12-5	29-4	12-5
5	36-3	12-4	82-5	82-5	96-2	12-5	12-5	12-5	29-4
6	12-2	61-3	17-4	17-4	17-4	81-2	27-5	27-5	27-5
7	12-2	25-4	42-3	17-4	17-4	27-5	83-2	27-5	27-5
8	12-2	25-4	17-4	42-3	17-4	27-5	27-5	83-2	27-5
9	12-2	25-4	17-4	17-4	42-3	27-5	27-5	27-5	83-2
R_j	22.5	.338	.227	.227	2.06	.147	.166	.166	.201

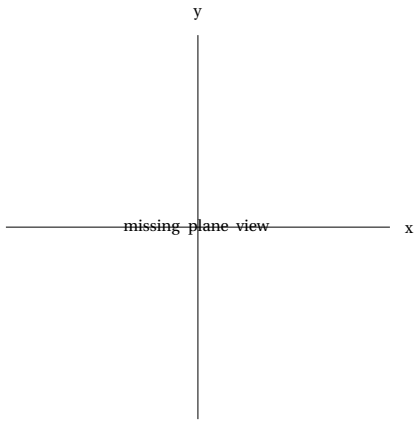
Out[387]=

|R|/|q|	$\bar{k}$
26.016	.9616

anisotropy	<



dB[Φ_j/Φ_1]



In[388]:=
```
configureExperiment [dressing → False ,
 materialGeometry → {{"Pile ", .04, 1.01850, 0, 0, 0, 0, 1},
                     {"Sample ", 9, .5, 0, 0, 0, Sqrt[2/3] - 1/(2 Sqrt[6]), .12},
                     {"Control ", 3, .5, 0, 1/Sqrt[3], 0, -1/(2 Sqrt[6]), .12},
                     {"Detector ", 3, .5, 0, -1/(2 Sqrt[3]), 1/2, -1/(2 Sqrt[6]), .12},
                     {"ExtraSource ", 3, .5, 1, -1/(2 Sqrt[3]), -1/2, -1/(2 Sqrt[6]), .12}}];
buildExperiment [graphics → False]
runExperiment []
reportExperiment []
```

simplify : Simplifying Pij 10 seconds

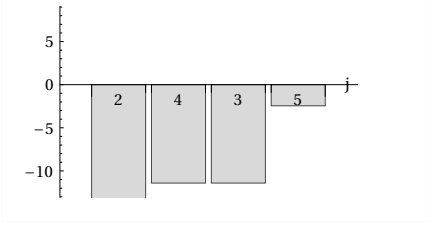
Out[389]= 0.268697

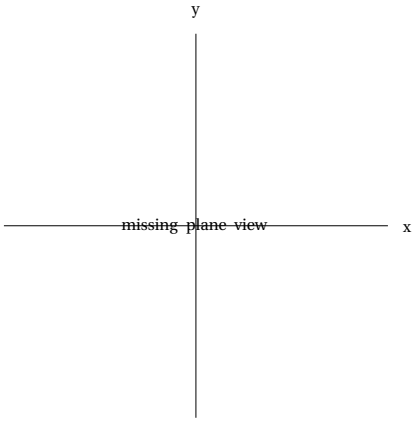
$P_{i\backslash j}$	1	2	3	4	5
1	98-2	70-4	47-4	47-4	47-4
2	18-3	98-2	40-5	40-5	40-5
3	36-3	12-4	96-2	82-5	82-5
4	36-3	12-4	82-5	96-2	82-5
5	36-3	12-4	82-5	82-5	96-2
R_j	28.5	.401	.269	.269	2.12
Φ_j	171.1	6.163	12.37	12.37	97.39

Out[390]=

|R|/|q|	$\bar{k}$
31.529	.9683

anisotropy	<





7.2. Linear solve

In[393]:=
```
configureExperiment []
```

In[394]:=
```
buildExperiment [toPerturb → {Σ2}, graphics → False , internals → False]
```

simplify : Simplifying Pij 10 seconds

In[395]:=
```
runExperiment []
```

Out[395]= 0.660061

Although the determinant is small, the solution is accurate:

```
In[172]:= With[
           {matrixVector1 =
              Echo[observable /. Options @ runExperiment /. Options @ runExperiment /.
                Hold[Part][Hold[Apply][LinearSolve , matrixVector_ ], 4] → matrixVector ]},
           With[{solution = LinearSolve @@ matrixVector1 },
            Print @ solution ;
            Print @ Det @ First @ matrixVector1 ;
            Print[First @ matrixVector1 .solution ];
            Print @ Last @ matrixVector1 ;
            Print[First @ matrixVector1 .solution - Last @ matrixVector1 ]
           ]]
```

```
» {{{-0.0169847 , 0.0181974 , 0.0181974 , 0.0181974 , 0.0181974 , 0.158509 ,
     0.158509 , 0.158509 , 0.158509 }, {0.00483746 , -0.522627 , 0.000407961 ,
     0.000407961 , 0.000407961 , 0.0277432 , 0.00191543 , 0.00191543 , 0.00191543 },
    {0.00483746 , 0.000407961 , -0.522627 , 0.000407961 , 0.000407961 , 0.00191543 ,
     0.0277432 , 0.00191543 , 0.00191543 }, {0.00483746 , 0.000407961 , 0.000407961 ,
     -0.522627 , 0.000407961 , 0.00191543 , 0.00191543 , 0.0277432 , 0.00191543 },
    {0.00483746 , 0.000407961 , 0.000407961 , 0.000407961 , -0.522627 , 0.00191543 ,
     0.00191543 , 0.00191543 , 0.0277432 }, {0.00403372 , 0.00265583 , 0.000183362 ,
     0.000183362 , 0.000183362 , -0.176489 , 0.000996799 , 0.000996799 , 0.000996799 },
    {0.00403372 , 0.000183362 , 0.00265583 , 0.000183362 , 0.000183362 , 0.000996799 ,
     -0.176489 , 0.000996799 , 0.000996799 }, {0.00403372 , 0.000183362 , 0.000183362 ,
     0.00265583 , 0.000183362 , 0.000996799 , 0.000996799 , -0.176489 , 0.000996799 },
    {0.00403372 , 0.000183362 , 0.000183362 , 0.000183362 , 0.00265583 ,
     0.000996799 , 0.000996799 , 0.000996799 , -0.176489 }},
   {-0.0363948 , -0.000815923 , -0.000815923 , -0.000815923 , -0.954746 ,
    -0.000366724 , -0.000366724 , -0.000366724 , -0.00531166 }}

  {60.8862 , 0.660061 , 0.660061 , 0.660061 , 2.48653 , 1.43211 , 1.43211 , 1.43211 , 1.48541 }

  -9.61206 × 10^-8

  {-0.0363948 , -0.000815923 , -0.000815923 , -0.000815923 ,
   -0.954746 , -0.000366724 , -0.000366724 , -0.000366724 , -0.00531166 }

  {-0.0363948 , -0.000815923 , -0.000815923 , -0.000815923 ,
   -0.954746 , -0.000366724 , -0.000366724 , -0.000366724 , -0.00531166 }

  {1.38778 × 10^-17, -7.00395 × 10^-17, -5.26922 × 10^-17, -2.79724 × 10^-17,
   -2.22045 × 10^-16, 3.18755 × 10^-17, 2.92735 × 10^-17, 5.44269 × 10^-17, 1.249 × 10^-16}
```

LinearSolve warns when it runs on a bad case (see Mathematica documentation of LinearSolve) so I won't double check.

7.3. Dressing effects

Very thin dressing

```
In[397]:= configureExperiment [dressing → True]
```

```
In[398]:= Drop[threadRule [r][Table[.1201 , {9}]], 5]
Out[398]= {r6 → 0.1201 , r7 → 0.1201 , r8 → 0.1201 , r9 → 0.1201 }

In[399]:= buildExperiment [Sequence @@ Drop[threadRule [r][Table[.1201 , {9}]], 5]]
  ••• simplify : Simplifying  Pij 10 seconds

In[400]:= runExperiment []
Out[400]= 3.83799
```

No dressing

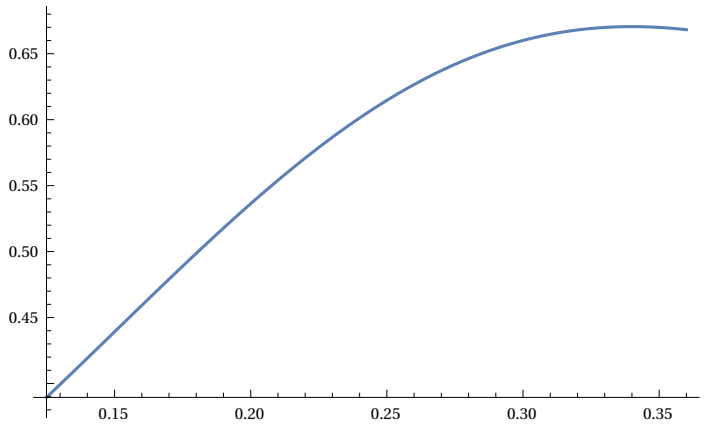
```
In[401]:= configureExperiment [dressing → False]

In[402]:= buildExperiment []
  ••• simplify : Simplifying  Pij 10 seconds

In[403]:= runExperiment []
Out[403]= 3.85203
```

Optimal dressing thickness

Perturb only one dressing radius:

```
In[ ]:= configureExperiment [dressing → True]
        buildExperiment [graphics → False , internals → False , toPerturb → {r6(*,r7,r8,r9*)}]

In[ ]:= r6 /. Options @ runExperiment
Out[ ]= 0.3

In[ ]:= runExperiment [r6 → .32]
Out[ ]= 0.668004

In[ ]:= Plot[runExperiment [r6 → r6], {r6 , .125 , .36}]
```

Out[ ]=



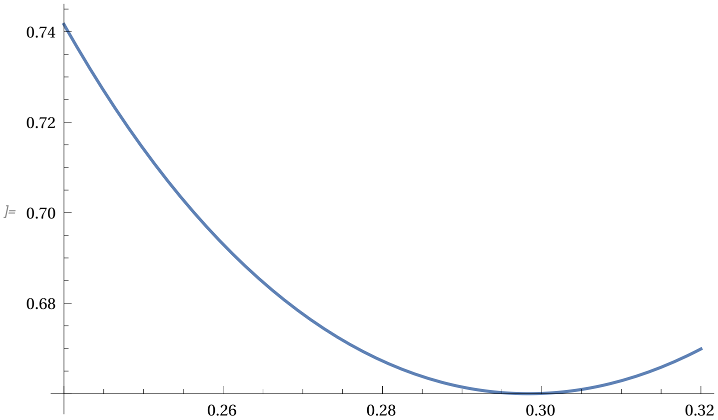
Perturb all dressing radii together:

In[∘]:=
```
buildExperiment [graphics → False , internals → False , toPerturb → {r6, r7, r8, r9}]
```

The stationary point, approximately 0.3, is the optimal dressing thickness :

In[∘]:=
```
Plot[runExperiment [Sequence @@ Thread [{r6, r7, r8, r9} → Table [r6 , 4]]], {r6 , .24 , .32}]
```

Out[∘]=



7.4. Anisotropy

In[∘]:=
```
configureExperiment []
```

In[∘]:=
```
buildExperiment [toPerturb → {}, dressing → True]
```

```
Simplifying  Pij  10  seconds
```

By symmetry, the anisotropy should be and is almost zero:

In[∘]:=
```
Chop @ MatrixForm [anisotropyMatrix /. Options @ runExperiment ]
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

Examine the anisotropy if the spheres are all translated along the z-axis to the x,y plane, that is to replace all z_i by zero. Also, as there is less room in plane, the dressing radius must be reduced to avoid error normalization error:

In[404]:=
```
configureExperiment []
buildExperiment [z2 → 0, z3 → 0, z4 → 0, z5 → 0, r6 → .2, r7 → .2, r8 → .2, r9 → .2]
runExperiment []
```

```
••• simplify : Simplifying  Pij  10  seconds
```

Out[406]=
```
0.106335
```

α_{ij} is symmetric because division by P_{ij} eliminates the anisotropic factor $V_i\,\Sigma_i$. α_{ij} can be non zero only if it represents a region appearing on more than one path: the pile 1 or some dressing (6,7,8,9). Casually, α_{16} is zero because each sphere other than 2 has symmetric position with respect to sphere 2:

```
In[ ]:= NumberForm [#, {4, 3}] &@ MatrixForm [Chop[anisotropyMatrix /. Options @ runExperiment ]]
```

Out[ ]//NumberForm=

$$\begin{pmatrix} 0 & 0 & 0.537 & 0.537 & 0.537 & 0 & 0.891 & 0.891 & 0.891 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.537 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.537 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.537 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.891 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.891 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.891 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
In[ ]:= configureExperiment []
        buildExperiment [x2 → .05, z2 → 0, z3 → 0, z4 → 0, z5 → 0, r6 → .2, r7 → .2, r8 → .2, r9 → .2]
        runExperiment []

      Simplifying  Pij  10  seconds

Out[ ]= 0.0955946
```

α_{1j} is greater for dressing (j=6,7,8,9) than interior (j=2,3,4,5), because the distance differences between the 3 possible axes are relatively greater:

```
In[ ]:= First[anisotropyMatrix /. Options @ runExperiment ]

Out[ ]= {0, 0.126121 , 0.622854 , 0.493664 , 0.493664 , 0.301765 , 1.02222 , 0.819358 , 0.819358 }
```

Probably the model should not be used for sphere positions such that α is greater than 1/10, so this quantity is reported as "anisotropy":

```
In[ ]:= Max @ Rest @ First[anisotropyMatrix /. Options @ runExperiment ]

Out[ ]= 1.02222
```

Constraining anisotropy is also a cheap way to avoid contact without watching the geometry.

8. Perturbation experiments

8.1. Open loop

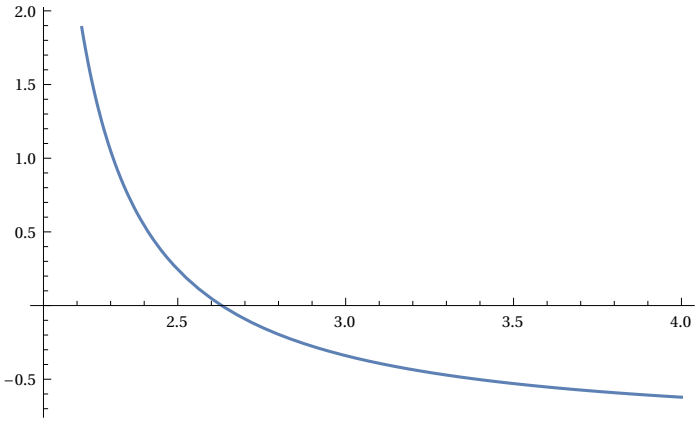
8.2. Closed loop

Constant detection rate curves on open loop ContourPlot are equivalent to closed loop but for more precise results closed loop requires solving a constant detection rate equation.

```
In[411]:= configureExperiment []

In[412]:= buildExperiment [toPerturb → {Σ2}, graphics → False , internals → False]
          runExperiment []

        ••• simplify : Simplifying  Pij  10  seconds

Out[413]= 0.660061

In[427]:= s2 =.
```

In[414]:=
```
Plot[runExperiment [Σ2 → s2] - 1, {s2, 2.1, 4}]
```

Out[414]=



In[415]:=
```
runExperiment [Σ2 → s2] - 1 /. s2 → 3
```

Out[415]=
```
-0.339939
```

Different methods are available:

In[428]:=
```
FindRoot [runExperiment [Σ2 → s2] - 1, {s2, 2.5}, Method → "Newton "] // Timing
```

Out[428]=
```
{0.016333 , {s2 → 2.63086 }}
```

In[429]:=
```
FindRoot [runExperiment [Σ2 → s2] - 1, {s2, 2.3, 3}, Method → "Brent "] // Timing
```

Out[429]=
```
{0.010725 , {s2 → 2.63086 }}
```

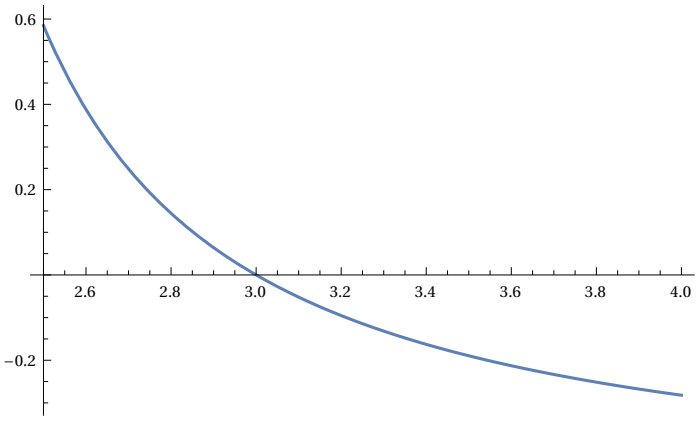
In[430]:=
```
bisection [runExperiment [Σ2 → s2] - 1, {s2, 2.3, 3}] // Timing
```

Out[430]=
```
{0.038655 , {s2 → 2.63086 }}
```

In[ ]:=
```
buildExperiment [toPerturb → {Σ2, k3}, graphics → False, internals → False]
```

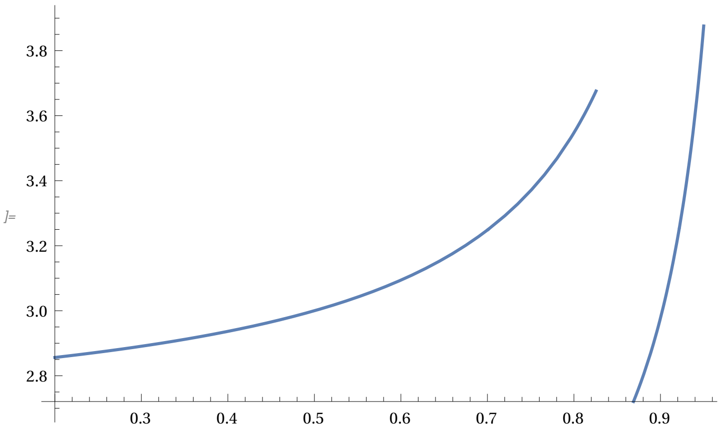
In[ ]:=
```
With[{R4 = runExperiment [], k3 = .5},
 Plot[-R4 + runExperiment [Σ2 → s2, k3 → k3], {s2, 2.5, 4}]]
```

Out[ ]=



```
In[ ]:= Clear @ s2fromk3 ;
        s2fromk3 @ k3_ :=
         With[{R4 = runExperiment []},
          s2 /. bisection [- R4 + runExperiment [Σ2 → s2 , k3 → k3], {s2 , 2.5 , 4}]]

In[ ]:= Plot[s2fromk3 @ k3 , {k3 , .2 , .95}]
```

Out[ ]=



The right branch is erroneous:

```
In[ ]:= s2fromk3 @ .9
```

LinearSolve : Result for LinearSolve of badly conditioned matrix {{-0.016961 , 0.0182474 , 0.0327553 , 0.0181974 , 0.0181974 , 0.158509 , 0.158509 , 0.158509 , 0.158509 }, ≪7 ≫, {0.00403372 , ≪7≫, -0.176479 }} may contain significant numerical errors .

LinearSolve : Result for LinearSolve of badly conditioned matrix {{-0.016961 , 0.0182474 , 0.0327553 , 0.0181974 , 0.0181974 , 0.158509 , 0.158509 , 0.158509 , 0.158509 }, ≪7 ≫, {0.00403372 , ≪7≫, -0.176479 }} may contain significant numerical errors .

LinearSolve : Result for LinearSolve of badly conditioned matrix {{-0.016961 , 0.0182474 , 0.0327553 , 0.0181974 , 0.0181974 , 0.158509 , 0.158509 , 0.158509 , 0.158509 }, ≪7 ≫, {0.00403372 , ≪7≫, -0.176479 }} may contain significant numerical errors .

General : Further output of LinearSolve ::luc will be suppressed during this calculation .

```
Out[ ]= 2.97712

In[ ]:= Clear @ s2fromk3 ;
        s2fromk3 @ k3_ :=
         With[{R4 = runExperiment []},
          s2 /. FindRoot [- R4 + runExperiment [Σ2 → s2 , k3 → k3], {s2 , 3}]]

In[ ]:= Plot[s2fromk3 @ k3 , {k3 , .2 , .95}]

Out[ ]= $Aborted
```

9. Interpretation

Explain experimental results by reasoning and with partly analytic solutions.

9.1. Spontaneous source perturbations

```
In[ ]:= configureExperiment [toPerturb → {q2, q5}]
```

```
In[ ]:= buildExperiment [graphics → False , internals → False , method → Inverse ]
```

```
Solving

Replacing

Simplifying
```

```
In[ ]:= Options @ runExperiment
```

```
Out[ ]= {toPerturb → {q2, q5}, q2 → 0, q5 → 1,
  stochasticMatrixDiagonal → {0.96516 , 0.954746 , 0.954746 , 0.954746 , 0.954746 ,
    0.808553 , 0.808553 , 0.808553 , 0.808553 }, observable → 0.660061 q2 + 0.660061 q5}
```

The general solution is a linear function of the perturbed parameters and the coefficients are equal by symmetry:

```
In[ ]:= observable /. Options @ runExperiment
```

```
Out[ ]= 0.660061 q2 + 0.660061 q5
```

```
In[ ]:= observable //. Options @ runExperiment
```

```
Out[ ]= 0.660061
```

```
In[ ]:= runExperiment []
```

```
Out[ ]= 0.660061
```

9.2. Multiplication factor perturbations

Open loop

```
In[431]:= configureExperiment [toPerturb → {k2}]
```

```
In[432]:= buildExperiment [graphics → False , internals → True , method → Inverse ]
```

```
••• simplify : Simplifying  Pij  10 seconds

Solving

Replacing

••• simplify : Simplifying  after replacing  500 seconds
```

```
In[433]:= First /@ Options @ runExperiment
```

```
Out[433]= {toPerturb , k2, detector , dressing , stochasticMatrix ,
  reciprocitySUTMatrix , anisotropyMatrix , stochasticMatrixDiagonal ,
  solution , observable , fluxes , amplificationMultiplication }
```

Homographic non-linearity:

In[434]:=
```
solution /. Options @ runExperiment
```

Out[434]=
$$\left\{R_1 \to \frac{46.8482 - 44.7236\ k_2}{0.902167 - 1.\ k_2},\ R_2 \to \frac{0.265455}{0.902167 - 1.\ k_2},\ R_3 \to \frac{0.507877 - 0.484845\ k_2}{0.902167 - 1.\ k_2},\right.$$
$$R_4 \to \frac{0.507877 - 0.484845\ k_2}{0.902167 - 1.\ k_2},\ R_5 \to \frac{2.15566 - 2.31131\ k_2}{0.902167 - 1.\ k_2},\ R_6 \to \frac{1.09485 - 1.0378\ k_2}{0.902167 - 1.\ k_2},$$
$$\left.R_7 \to \frac{1.10192 - 1.05195\ k_2}{0.902167 - 1.\ k_2},\ R_8 \to \frac{1.10192 - 1.05195\ k_2}{0.902167 - 1.\ k_2},\ R_9 \to \frac{1.15001 - 1.10525\ k_2}{0.902167 - 1.\ k_2}\right\}$$

In[435]:=
```
%[[4]] //. Options @ runExperiment
```

Out[435]=
$$R_4 \to 0.660061$$

In[436]:=
```
runExperiment []
```

Out[436]=
```
0.660061
```

Closed loop

In[437]:=
```
configureExperiment []
```

In[438]:=
```
buildExperiment [toPerturb → {k2, k3} , graphics → False , internals → True ,
 method → Inverse ]
```

```
••• simplify : Simplifying  Pij  10  seconds
Solving
Replacing
••• simplify : Simplifying  after  replacing  500  seconds
```

Family of homographic functions with a critical limit:

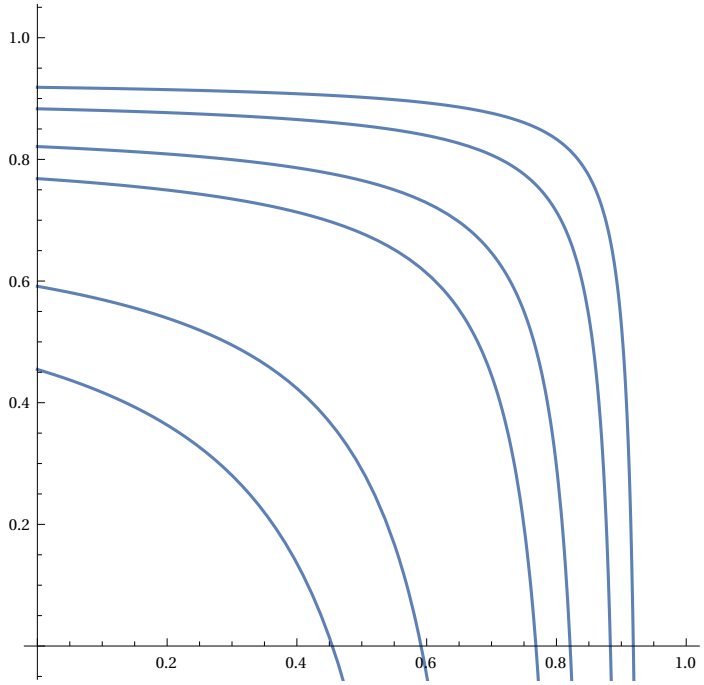
In[439]:=
```
With[
 {solution1 =
   Echo[
    Simplify @ Part[solution /. Options @ runExperiment ,
        detector /. Options @ runExperiment ] /. x_ / y_ :> monomialCollect [{k2, k3}][x / y]],
 With[
  {NSolve1 = Echo @ only @ NSolve [solution1 /. Rule → Equal , k3] /.
     x_ / y_ :> monomialCollect [{k2}][x / y],
   toPlot2 = (*the critical relation *)
    Echo[k3 /. only @ NSolve [Denominator @ Last @ solution1 == 0, k3]] /. k2 → k2
  },
  With[{toPlot1 = k3 /. NSolve1 /. R4 → # & /@ {.55 , .6 , .8 , 1. , 2} /. k2 → k2},
   Show[plotBranch [#, {k2 , 0 , .95}, 1 , PlotRange → Full] & /@ Prepend [toPlot1 , toPlot2 ],
    AxesOrigin → {0, 0}, PlotRange → {{0, 1}, {0, 1}}, AspectRatio → 1]
  ]
 ]]
```

» $R_4 \to \dfrac{0.420405 - 0.40134\ k_2 - 0.40134\ k_3 + 0.383139\ k_2\ k_3}{0.856654 - 0.932655\ k_2 - 0.932655\ k_3 + 1.\ k_2\ k_3}$

» $\{k_3 \to (2.26657 \times 10^{-11}\,(8.80111 \times 10^{32} - 8.40197 \times 10^{32}\ k_2 - 1.79339 \times 10^{33}\ R_4 + 1.95249 \times 10^{33}\ k_2\ R_4))\ /$
$(1.90437 \times 10^{22} - 1.818 \times 10^{22}\ k_2 - 4.42548 \times 10^{22}\ R_4 + 4.74503 \times 10^{22}\ k_2\ R_4)\}$

» $\dfrac{-0.856654 + 0.932655\ k_2}{-0.932655 + 1.\ k_2}$

Out[439]=



Non-linearity is stronger in the critical limit (this is just a scale and window effect since all hyperbola are homothetic).

9.3. Cross section perturbations

Open loop

In[440]:=
```
configureExperiment [toPerturb → {Σ2}]
```

In[441]:=
```
buildExperiment [graphics → False , internals → False , method → LinearSolve ];
runExperiment []
```

··· simplify : Simplifying Pij 10 seconds

Out[441]= 0.660061

In[442]:=
```
Timing @ buildExperiment [graphics → False , internals → False , method → Inverse ]
```

··· simplify : Simplifying Pij 10 seconds

Solving

Replacing

··· simplify : Simplifying after replacing 70 seconds

Out[442]= {70.9324 , Null}

In[443]:= `First /@ Options @ runExperiment`

Out[443]= {toPerturb , Σ_2, stochasticMatrixDiagonal , observable }

In[444]:= `ByteCount /@ Options @ runExperiment`

Out[444]= {224 , 192 , 4800 , 40 304 }

In[445]:= `runExperiment []`

Out[445]= 0.660061

In[447]:= `Short[#, 15] &[observable /. Options @ runExperiment ]`

Out[447]//Short=
$$(\ll 45 \gg + \text{expM}[0.0072 + 0.12\ \Sigma_2]\ (-2.55669 \times 10^{-9} - 2.408 \times 10^{-9}\ \text{xExpM1}[0.12\ \Sigma_2] - 4.27899 \times 10^{-10}\ \text{xExpM1}[0.12\ \Sigma_2]^2 - 3.06848 \times 10^{-11}\ \text{xExpM1}[0.12\ \Sigma_2]^3 - 1.00509 \times 10^{-12} \ll 1 \gg^4 - 1.34113 \times 10^{-14}\ \text{xExpM1}[0.12\ \Sigma_2]^5 - 3.28535 \times 10^{-17}\ \text{xExpM1}[0.12\ \Sigma_2]^6 + 2.25513 \times 10^{-35}\ \text{xExpM1}[0.12\ \Sigma_2]^7)) /$$
$$(2.30653 \times 10^{-7} + \ll 43 \gg + \text{expM}[0.0072 + 0.12\ \Sigma_2]\ (1.21191 \times 10^{-8} + 1.0187 \times 10^{-8}\ \text{xExpM1}[0.12\ \Sigma_2] + 1.46116 \times 10^{-9}\ \text{xExpM1}[0.12\ \Sigma_2]^2 + 8.25976 \times 10^{-11}\ \text{xExpM1}[0.12\ \Sigma_2]^3 + \ll 24 \gg \ll 1 \gg^4 + 2.12028 \times 10^{-14} \ll 1 \gg^5 + 8.10897 \times 10^{-17}\ \text{xExpM1}[0.12\ \Sigma_2]^6 + 1.0204 \times 10^{-19}\ \text{xExpM1}[0.12\ \Sigma_2]^7 + 3.29125 \times 10^{-35}\ \text{xExpM1}[0.12\ \Sigma_2]^8))$$

In[448]:= `observable /. Options @ runExperiment /. Options @ runExperiment`

Out[448]= 0.660061

Closed loop

In[ ]:= `configureExperiment [toPerturb → {Σ2, k3}]`

In[ ]:=
```
Timing @ buildExperiment [graphics → False , internals → False , method → Inverse ]
runExperiment []
```

```
Solving
Replacing
Simplifying
```

Out[ ]= {127.517 , Null}

Out[ ]= 0.660061

In[ ]:=
```
First /@ Options @ runExperiment
ByteCount /@ Options @ runExperiment
```

Out[ ]= {toPerturb , Σ_2, k_3, stochasticMatrixDiagonal , observable }

Out[ ]= {352 , 192 , 192 , 4872 , 81 208 }

In[ ]:=
```
Short[#, 15] &[observable /. Options @ runExperiment ]
```

Out[ ]//Short=
```
(≪72≫ + expM[0.0232 + 0.12 Σ2]
      (≪12≫ + k3 (1.43673 × 10^-10 + 1.63001 × 10^-10 xExpM1[0.12 Σ2] + 4.59457 × 10^-11 xExpM1[0.12 Σ2]^2 +
            4.73831 × 10^-12 xExpM1[≪1≫]^3 + ≪24≫ ≪1≫^4 + 2.53517 × 10^-15 xExpM1[0.12 Σ2]^5 -
            1.10899 × 10^-17 xExpM1[0.12 Σ2]^6 + 3.00927 × 10^-36 xExpM1[0.12 Σ2]^7))) /
    (4.17312 × 10^-7 + ≪77≫ + expM[0.0232 + 0.12 Σ2] (6.73509 × 10^-10 + ≪11≫ +
          k3 (- 6.02505 × 10^-10 - 1.871 × 10^-10 xExpM1[0.12 Σ2] + 1.5965 × 10^-10 xExpM1[0.12 Σ2]^2 +
            1.49002 × 10^-11 xExpM1[≪5≫ Σ2]^3 + ≪24≫ ≪1≫ + ≪23≫ ≪1≫^5 + 3.64127 × 10^-18
             xExpM1[0.12 Σ2]^6 + 1.58315 × 10^-20 xExpM1[0.12 Σ2]^7 + 1.50463 × 10^-36 xExpM1[0.12 Σ2]^8)))
```

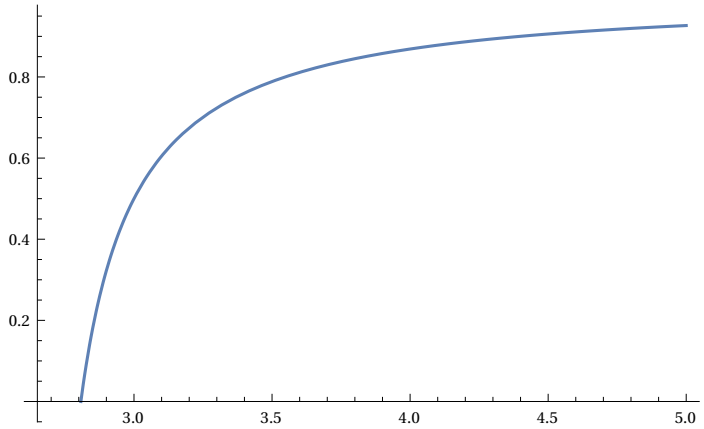
In[ ]:=
```
k3 /. Options @ runExperiment
```

Out[ ]=
```
0.5
```

In[ ]:=
```
Clear @ k3froms2 ;
k3froms2 @ s2_ :=
 With[{R4 = runExperiment []},
  k3 /. FindRoot [- R4 + runExperiment [Σ2 → s2 , k3 → k3], {k3 , .5}]]
```

In[ ]:=
```
Plot[k3froms2 @ s2 , {s2 , 2.65 , 5}]
```

Out[ ]=



In[ ]:=
```
configureExperiment [toPerturb → {Σ2, Σ3}]
```

In[ ]:=
```
Timing @ buildExperiment [graphics → False , internals → False , method → Inverse ]
runExperiment []
```

```
Solving

Replacing

Simplifying
```

Out[ ]=
```
{430.234 , Null}
```

Out[ ]=
```
0.660061
```

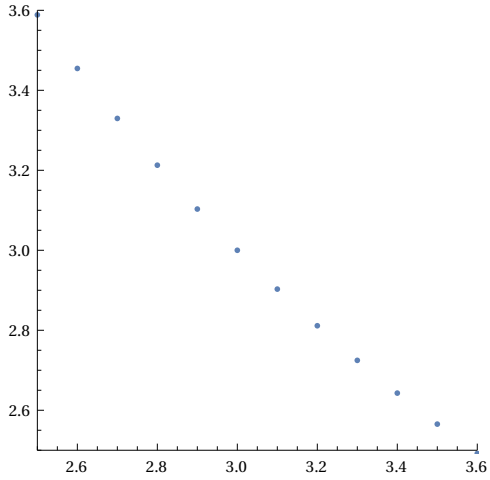
```
In[ ]:= First /@ Options @ runExperiment
        ByteCount /@ Options @ runExperiment

Out[ ]= {toPerturb , Σ2, Σ3, stochasticMatrixDiagonal  , observable }

Out[ ]= {352 , 192 , 192 , 9640 , 1 402 528 }

In[ ]:= Short[#, 20] &[observable /. Options @ runExperiment ]

Out[ ]//Short= (≪889≫ + expM[≪21≫ + 0.12 Σ2] ≪1≫ +
                   expM[0.0232 + 0.12 Σ2] (≪158≫ + expM[≪1≫] ≪1≫ + expM[0.0232 + 0.12 Σ3]
                        (-9.69102 × 10^-14 - 2.75477 × 10^-18 xExpM1[0.12 Σ2]^5 + ≪13≫ + xExpM1[0.12 Σ2]
                            (-1.12446 × 10^-13 - 1.20195 × 10^-13 xExpM1[0.12 Σ3] - 3.04389 × 10^-14 xExpM1[0.12 Σ3]^2 -
                               2.44818 × 10^-15 xExpM1[0.12 Σ3]^3 - 5.84832 × 10^-17 xExpM1[0.12 Σ3]^4 -
                               1.20371 × 10^-35 xExpM1[0.12 Σ3]^5)))) /
              (4.18133 × 10^-7 + ≪908≫ + expM[0.0232 + 0.12 Σ2] (6.74833 × 10^-10 + ≪165≫ +
                     expM[0.0232 + 0.12 Σ3] (1.08913 × 10^-12 - 2.82119 × 10^-37 xExpM1[0.12 Σ2]^7 +
                         ≪13≫ + xExpM1[0.12 Σ2] (8.79193 × 10^-13 + 4.44639 × 10^-13 xExpM1[0.12 Σ3] -
                             3.84371 × 10^-14 xExpM1[0.12 Σ3]^2 - 2.21619 × 10^-15 xExpM1[0.12 Σ3]^3 + 1.40045 × 10^-17
                              xExpM1[0.12 Σ3]^4 + 6.7477 × 10^-20 xExpM1[0.12 Σ3]^5 - 4.5139 × 10^-36 xExpM1[0.12 Σ3]^6))))
```

For bisection, the constant R curve can only be obtained on a square where the function is defined everywhere.

```
In[ ]:= runExperiment []

Out[ ]= 0.660061

In[ ]:= runExperiment [Σ2 → 2.48 , Σ3 → 2.48]

Out[ ]= 37.011

In[ ]:= runExperiment [Σ2 → 2.48 , Σ3 → 3.61]

Out[ ]= 0.663205

In[ ]:= % > runExperiment []

Out[ ]= True

In[ ]:= Clear @ s3froms2 ;
        s3froms2 @ s2_ :=
         With[{R4 = runExperiment []},
          s3 /. bisection [-R4 + runExperiment [Σ2 → s2 , Σ3 → s3], {s3 , 2.48 , 3.61}, AccuracyGoal → 6]]
```

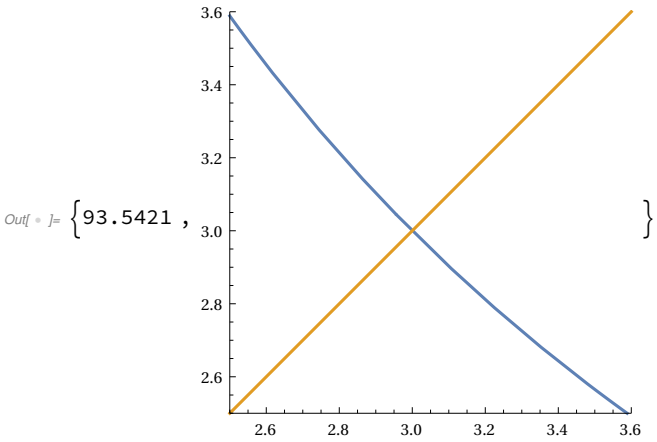
In[ ]:=
```
ListPlot[Table[{s2, s3froms2 @ s2}, {s2, 2.5, 3.6, .1}], PlotRange → {{2.5, 3.6}, {2.5, 3.6}},
 AspectRatio → 1, ImageSize → 250]
```

Out[ ]=



In[ ]:=
```
With[{min = 2.5, max = 3.6}, Plot[{s3froms2 @ s2, s2}, {s2, min, max},
   PlotRange → {{min, max}, {min, max}}, AspectRatio → 1, ImageSize → 230, PlotPoints → 10]] //
 Timing
```

Out[ ]=



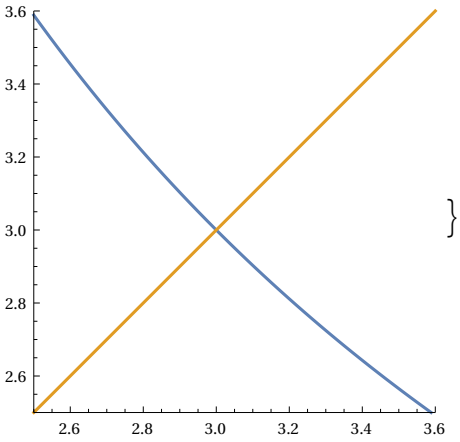
In[ ]:=
```
Clear @ s3froms2 ;
s3froms2 @ s2_ :=
 With[{R4 = runExperiment []},
  s3 /. FindRoot [- R4 + runExperiment [Σ2 → s2, Σ3 → s3], {s3, 3}]]
```

In[ ]:=
```
s3froms2 @ 3.
```

Out[ ]=
```
3.
```

FindRoot is faster than bisection:

```
In[ ]:= With[{min = 2.5, max = 3.6}, Plot[{s3froms2 @ s2, s2}, {s2, min, max},
          PlotRange → {{min, max}, {min, max}}, AspectRatio → 1, ImageSize → 230]] // Timing
```

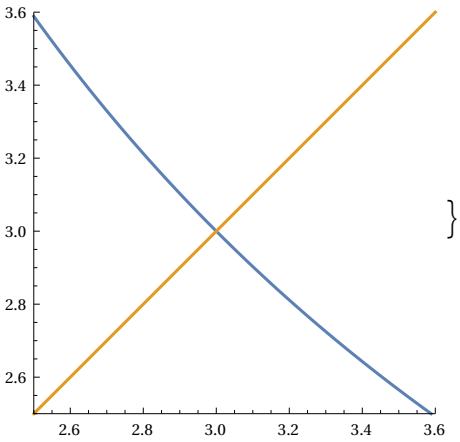



```
Out[ ]= {67.7081 , }
```

```
In[ ]:= buildExperiment [graphics → False , internals → False , method → LinearSolve ]
```

Numeric is faster than symbolic:

```
In[ ]:= With[{min = 2.5, max = 3.6}, Plot[{s3froms2 @ s2, s2}, {s2, min, max},
          PlotRange → {{min, max}, {min, max}}, AspectRatio → 1, ImageSize → 230]] // Timing
```




```
Out[ ]= {1.83829 , }
```

The convexity of the graph implies that R is not constant on the line $\Sigma_2 + \Sigma_3 == 6$, it is minimum at the symmetric point. It also implies that a decrease of one cross section is balanced by an opposite sign and *larger* magnitude increase of the other. Overall, the same amount of absorbing material is more efficient if evenly distributed. This also implies *self-shielding*: that the flux decreases relatively with the local cross section. Show this numerically.

```
In[ ]:= configureExperiment []
```

```
In[ ]:= buildExperiment [toPerturb → {Σ2, Σ3}, graphics → False , internals → True ,
          method → LinearSolve , dressing → True]
```

```
In[ ]:= runExperiment []
```

```
Out[ ]= 0.660061
```

When Σ_2 increases, Φ_2 decreases:

In[ ]:=
```
Table[
  With[{s3 = s3froms2 @ s2},
   runExperiment [Σ2 → s2, Σ3 → s3];
   Flatten @{{s2, s3}, Extract[fluxes /. Options @ reportExperiment , {{1}, {2}, {3}}]}],
 {s2, 2.6, 3.4, .1}] // TableForm [#, TableHeadings → {None, {Σ2, Σ3, Φ1, Φ2, Φ3}}] &
```

Out[ ]//TableForm=

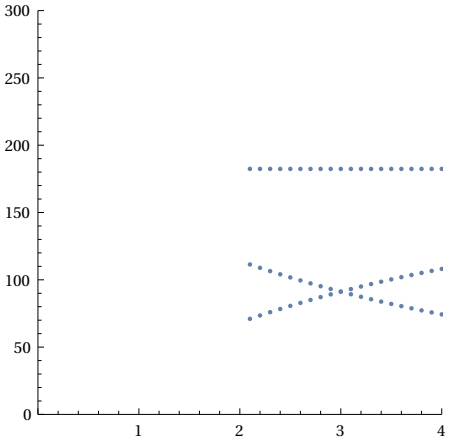
Σ_2	Σ_3	Φ_1	Φ_2	Φ_3
2.6	3.45464	365.92	31.8673	28.8111
2.7	3.32967	365.918	31.4929	29.2383
2.8	3.21275	365.917	31.1231	29.6439
2.9	3.10309	365.917	30.7578	30.0296
3.	3.	365.917	30.397	30.397
3.1	2.90291	365.917	30.0406	30.7473
3.2	2.81131	365.917	29.6885	31.0816
3.3	2.72476	365.918	29.3407	31.4009
3.4	2.64289	365.919	28.997	31.7062

In[ ]:=
```
Table[With[{s3 = s3froms2 @ s2},
  runExperiment [Σ2 → s2, Σ3 → s3];
  {s2, s3}.Extract[fluxes /. Options @ reportExperiment , {{2}, {3}}]], {s2, 2.6, 3.4, .1}]
```

Out[ ]= {182.387, 182.385, 182.383, 182.382, 182.382, 182.382, 182.383, 182.384, 182.386}

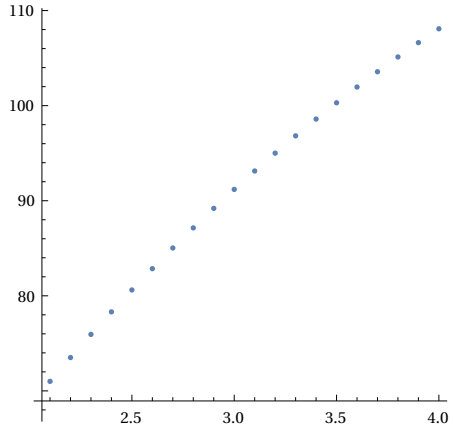
In[ ]:=
```
ListPlot[Flatten[#, 1] &@Table[With[{s3 = s3froms2 @ s2},
    runExperiment [Σ2 → s2, Σ3 → s3];
    {s2, #} & /@ {#1, #2, #1 + #2} &[s2 Part[fluxes /. Options @ reportExperiment , 2],
     s3 Part[fluxes /. Options @ reportExperiment , 3]]], {s2, 2.1, 4., .1}],
 AspectRatio → 1, ImageSize → 230, PlotRange → {{0, 4}, {0, 300}}]
```

Out[ ]=



```
In[ ]:= ListPlot[Table[With[{s3 = s3froms2 @ s2},
          runExperiment [Σ2 → s2, Σ3 → s3];
          {s2, s2 Part[fluxes /. Options @ reportExperiment , 2]}], {s2, 2.1, 4, .1}],
        AspectRatio → 1, ImageSize → 230]
```

Out[ ]=



10. Conclusions and possible extensions

The integral transport equation and the collision probability method

The integral form of the transport equation eliminates any differential operator (in some function or distribution space). This is simple.

The collision probability method is *pragmatic* insofar as it only accounts for actual events (collisions), not fluxes (although these can be determined later on). In particular, flux is not determined in any non reactive region, exactly where it is not needed. This agrees with the physical principle that we don't know if a particle exists until it has a collision.

Furthermore, the collision probability method eliminates by discretization any integral operator, although the integrals are pushed into the definition of collision probabilities.

Markov eigenvalue equation

The Markov eigenvalue equation {3} applies to any Markov chain with multiplicative scattering and spontaneous source. It can be used as a model for epidemic studies.

If all the P_{ij} are equal (to the inverse dimension) then the Markov eigenvalue equation has a uniform solution (see subsection 3.3). This is more generally true if the transposed stochastic matrix is also stochastic, because the uniform solution is then both right and left eigenvector.

Collision probability model

The collision probability model {15} applies to particles moving on straight lines over distances much greater than the collision scale, like neutrons in a nuclear fission reactor or photons in a photomultiplier (and such systems may be coupled). However, electrons rarely move on straight lines because they are sensitive to the electromagnetic field. This could work with electrons in the same conditions as required for electronic microscopy.

If the cross section field is invariant by exchange of any two regions, then $P_{ii} == P_{11}$, and for all $i \neq j$, $P_{ij} == P_{12}$. This might be possible in the regular tetrahedron geometry. The P_{ii} can be determined directly for simple shapes, see [1].

The collision probability defines a measure over space. When the target region grows, the probability increases but less than linearly because of flux depletion: this is the (spatial) self-shielding effect, that can be studied with the model. Normally thick regions should not be treated with the present model but just in case I have introduced a protection against non monotonous probability inconsistency, at the cost of breaking reciprocity (which is signaled by gray background in the probability table).

Finally, with the Inverse solution method, a fully symbolic (open loop) solution is obtained, based on the four basic operations and the exponential function. It is even readable is some cases. Even the path integrals giving the collision probabilities have been approximated by on a single path, which is equivalent to using the Beer-Lambert- Bouguer law. (Historically, it seems that

Pierre Bouguer missed the exponential function [4].)

Spherically symmetric configuration

Another geometry that can be obtained with spheres only is the "onion", occuring in particular in the first stages of an explosion. In this case, kinetics, including flight times, and inertial effects should be represented.

Simplified kinetics for oscillation experiments

The Markov eigenvalue equation {3} has a fixed point form $R == f[R]$ {6}. The iteration $R[0] == 0$, $R[k+1] == f[R[k]]$, $k \geq 0$ not only is the most natural solution method (slower than LinearSolve or Inverse), but also a kinetic model on its own, that could be used as a model for sample oscillation (periodic perturbation) experiments.

In particular, a delayed population group (as occurs for neutrons produced by nuclear fission) could be simply represented simply with delayed iteration (from k to k+p instead of k+1).

Physical measurement

If the relation between Σ_2, k_3 is known and k_3 is known, then Σ_2 can be determined from k_3 [5].

References

BussacReuss

[1] J. Bussac and P. Reuss, *Traité de neutronique*, Hermann, 1985.

stochastic matrix

[2] Stochastic matrix https://en.wikipedia.org/wiki/Stochastic_matrix

Lambert

[3] Beer-Lambert law https://en.wikipedia.org/wiki/Beer%E2%80%93Lambert_law

Bouguer

[4] P. Bouguer, *Essai d'optique sur la gradation de la lumière*, https://archive.org/details/UFIE003101_TO0324_PNI-2703_000000

Albarede

[5] P. Albarède, *Weighing operator perturbation from quasi-critical source system response*, arXiv:math-ph/0007026, 2000-2001.