Uncertainty paper

Principle

This page computes the errors introduced on volumes and concentrations introdiced by pipeting.

We'll use the law of propagation of uncertainty :

for $y = f(x_i)$ and $u(a)$ the uncertainty on a,

the uncertainty on y is given by :

$$u^2(y) = \sum_{i}^{N} \left(\frac{\partial f}{\partial x_i} \right)^2 u^2(x_i) + 2 \sum_{i}^{N-1} \sum_{j=i+1}^{N} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \text{cov}(x_i, x_j)$$

Dilution

Two possible dilution protocols :

1. add aliquotes to a given tube
2. create n different tubes

from these 2 approaches, we can compute the error on concentration for each sample.

In all cases, we consider that :

- error on volume by pipeting depends on the nominal value of the pipet system as well as on the pipeted volumes.
- the optimal pipet is always chosen
- error on the concentration of the stock solution measured by UV or NMR is 10%
- when working in one tube, there is an error at each aliquote due manipulation. It is considered to be 2.5%

concentration is given by $c = \frac{n}{v}$ where n is the number of mole, and v is the volume

thus $n = cv$

When pipeting v_1 of compound at c_1 and v_2 of compound at c_2, final solution is :

$v' = v_1 + v_2$ with compound at $c_p = \frac{c_1 v_1 + c_2 v_2}{v_1 + v_2}$

so it comes for uncertainty

$u^2(c_p) = (\frac{\partial c_p}{\partial c_1})^2 u^2(c_1)$

$$u^2(c_p) = (\frac{\partial c_p}{\partial\, c_1})^2 u^2(c_1) \;+ (\frac{\partial c_p}{\partial\, v_1})^2 u^2(v_1) \;+ (\frac{\partial c_p}{\partial c_2})^2 u^2(c_2) \;+ (\frac{\partial c_p}{\partial v_2})^2 u^2(v_2) + 2\,\frac{\partial c_p}{\partial\, c_1}\,\frac{\partial c_p}{\partial c_2}\,cov(c_1, c_2) + 2\,\frac{\partial c_p}{\partial v_1}\,\frac{\partial c_p}{\partial v_2}\,cov(v_1, v_2)$$

We can consider that

- $cov(c, v) = 0$
- $cov(v_1, v_2) \neq 0$ when the same pipet is used twice
- $cov(c_1, c_2) \neq 0$ when the same solution is used twice

Let's build this relations in the program :

```
var('v1 v2 c1 c2')
cp(c1, c2, v1, v2) = (c1*v1 + c2*v2)/(v1+v2)
cp
```

$$(c_1, c_2, v_1, v_2) \mapsto \frac{c_1 v_1 + c_2 v_2}{v_1 + v_2}$$

and

```
derivative(cp,c1)
```

$$(c_1, c_2, v_1, v_2) \mapsto \frac{v_1}{v_1 + v_2}$$

```
derivative(cp,v1)
```

$$(c_1, c_2, v_1, v_2) \mapsto \frac{c_1}{v_1 + v_2} - \frac{c_1 v_1 + c_2 v_2}{\left(v_1 + v_2\right)^2}$$

```
derivative(cp,c1)*derivative(cp,c2)
```

$$(c_1, c_2, v_1, v_2) \mapsto \frac{v_1 v_2}{\left(v_1 + v_2\right)^2}$$

We are now ready to compute the final uncertainty from all parameters.

We define a function `ucp1` for this

```
var("uc1, uc2, uv1,uv2, cc1c2, cv1v2")
ucp1(c1,c2,v1,v2,uc1, uc2,uv1,uv2,cc1c2, cv1v2) = sqrt(
(uv1*derivative(cp,v1))^2 + (uv2*derivative(cp,v2))^2 +
(uc1*derivative(cp,c1))^2 + (uc2*derivative(cp,c2))^2 +
2*derivative(cp,c2)*derivative(cp,c1)*cc1c2*uc1*uc2 +
2*derivative(cp,v2)*derivative(cp,v1)*cv1v2*uv1*uv2)
ucp1
```

$$(c_1, c_2, v_1, v_2, \mathrm{uc}_1, \mathrm{uc}_2, \mathrm{uv}_1, \mathrm{uv}_2, \mathrm{cc1c}_2, \mathrm{cv1v}_2) \mapsto \sqrt{2\,\mathrm{cv1v}_2\mathrm{uv}_1\mathrm{uv}_2\left(\frac{c_1}{v_1 + v_2} - \frac{c_1 v_1 + c_2 v_2}{\left(v_1 + v_2\right)^2}\right)\left(\frac{c_2}{v_1 + v_2} -\right.}$$

Let's also introduce the law for additivity (will be usefull for volumes)

```
def uadd(u1,u2,cov=0.0):
    "computes the uncertainty after addition"
    u =  sqrt(u1^2 + u2^2 + 2*cov*u1*u2)
    return u
```

And some constants used through out in the program :

trf_err : erreur on volume added when transfering from one vessel to another

conc_err : error on concentration when concentration obtained by UV measurement

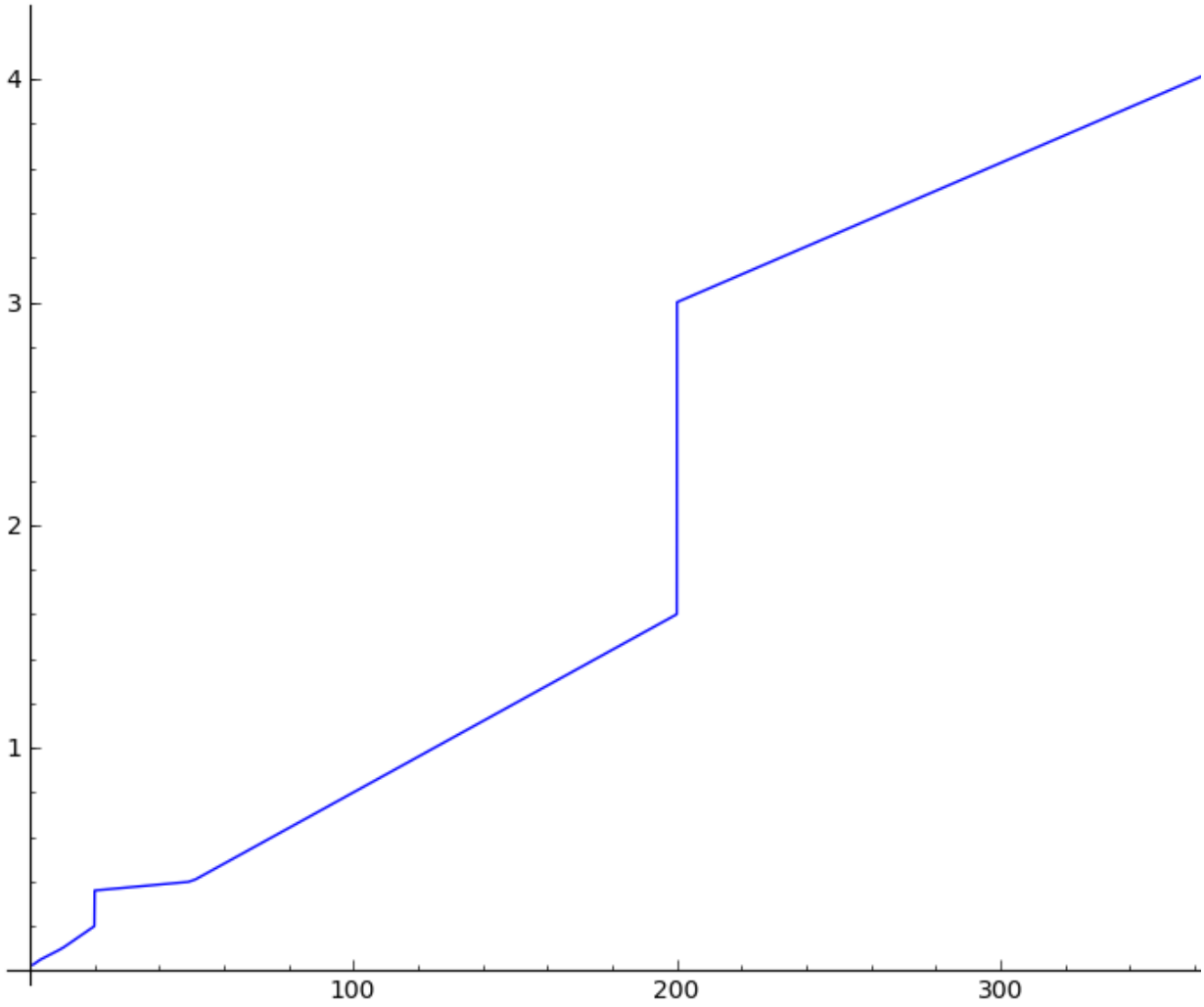
```
# cste definition
trf_err = 0.025    # 2.5% error at tube transfer
conc_err = 0.1     # 10%   error on concentration measurements
```

we have to implement numerically the function that computes the error on the volume

Errors based on the Gilson Specifications

```
def uncert_vol(vol):
    """
    returns the uncertainty of a pipeted volume - every thing in ul
    based (approximately) on the Gilson specifications
    """
    P = {}
    #name  ( (low, %err), (high, %err))  assume linear progression of
the error through the range
    P[2] = ((0.5,5), (2,1.5))
    P[10] = ((2,2), (10,1))
    P[20] = ((10,1),(20,1))
    P[50] = ((20,1.8),(50,0.8))    # pseudo pipet, to model the 20-50
range of the P100
    P[100] = ((50,0.8),(100,0.8))
    P[200] = ((100,0.8), (200,0.8))
    P[1000] = ((200,1.5), (1000,0.8))
    if vol == 0:
        return 0.0
    for v in (2, 10, 20, 50, 100, 200, 1000):    # go through pipets
        if vol<=v:                # found one
            mini,maxi = P[v]
            ermin = 0.01*mini[0]*mini[1]
            ermax = 0.01*maxi[0]*maxi[1]
            error = ermin + (ermax-ermin) * (vol-mini[0])/(1.0*maxi[0]-
mini[0])
            break
    return error
def ustr(A, uA, unit="uM"):
    "utility which returns formated string for uncertainty"
    return "%.2f %s +/- %.2f"%(A,unit,uA)
for v in (0.5, 10, 20, 30,50, 150, 500):
    print ustr(v, uncert_vol(v), "ul")
plot(uncert_vol,(1,400))
```

```
0.50 ul +/- 0.03
10.00 ul +/- 0.10
20.00 ul +/- 0.20
30.00 ul +/- 0.37
50.00 ul +/- 0.40
150.00 ul +/- 1.20
500.00 ul +/- 4.88
```



One remark

There are two problematic regions (confirmed by a careful reading of the specifications) :

- a weakness of the P200 between 20 and 50 μl,
- large jump when switching to the P1000

This implies that for certain volumes, one is better off by using twice the pipet just below :

```
print "using P1000 :   ", ustr(300, uncert_vol(300))
print "using 2 x P200 :", ustr(300,  2*uncert_vol(150))
print "using P100 :   ", ustr(30, uncert_vol(30))
print "using 2 x P20 :", ustr(30,  2*uncert_vol(15))
```

```
using P1000 :    300.00 uM +/- 3.62
using 2 x P200 : 300.00 uM +/- 2.40
using P100 :    30.00 uM +/- 0.37
using 2 x P20 : 30.00 uM +/- 0.30
```

Now we can implement a function which computes the complete uncertainty for a given pipeting :

```
def uncert_conc(c1, c2, v1, v2, uc1=None, uc2=None, uv1=None, uv2=None,
covc12=0.0, covv12=0.0):
    """
    returns the uncertainty on concentration upon dilution of vol v1 at
conc c1 into v2 at conc c2.
    ucx is optional and is uncertainty on concentration (5% default)
```

```
    uvx is optional and is uncertainty on volume
    covc12 is the covariance of uc1 and uc2 (1.0 if same solution)
    covv12 is the covariance of uv1 and uv2
    """
    if uc1 == None:
        uc1 = conc_err*c1     # assume std error on stock solution
    if uc2 == None:
        uc2 = conc_err*c2     # assume std error on stock solution
    if uv1 == None:
        uv1 = uncert_vol(v1)
    if uv2 == None:
        uv2 = uncert_vol(v2)
    u = ucp1(c1, c2, v1, v2, uc1, uc2, uv1, uv2, covc12, covv12)
    return u
```

and check it on a simple dilution :

(all concentrations are in μM and volumes in μL)

- stock solution : C1 = 1000
- pipeted volume : v1 = 10
- added to buffer volume : v2 = 490

```
## first example
C1 = 1000.0
V1 = 10.0
V2 = 490.0
print "%d ul of stock at %d uM, diluted into %d ul"%(V1, C1, V2)

VT = V1+V2
UVT = uadd(uncert_vol(V1),uncert_vol(V2))
print "final volume:", ustr(VT, UVT, unit="ul")

CT = cp(C1,0.0,V1,V2)
UCT = uncert_conc(C1,0.0,V1,V2,uv1=uncert_vol(V1),uv2=uncert_vol(V2))
print "final conc:", ustr(CT, UCT)
```

```
10 ul of stock at 1000 uM, diluted into 490 ul
final volume: 500.00 ul +/- 4.81
final conc: 20.00 uM +/- 2.02
```

A difficult remains here, as we have to consider correlated errors, otherwise some calculus will be false.

For insance, when mixing twice the same solution, as they are considered uncorrelated (i.e. 2 completely different preparation), the error will be reduced. Of course this is wrong if it is the same preparation.

eg. mixing equi volume of the same solution at 100 μl at 10% accuracy, leads to a 7% accuracy !

```
print uncert_conc(100.0,100.0,100.0,100.0, 10,10)
```

```
7.07106781186548
```

But not if we note that the errors on both conscentrations are indeed correlated, and state that $cov(c_1, c_2) = 1$

```
print uncert_conc(100.0,100.0,100.0,100.0,10,10,covc12=1.0)
```

```
10.0000000000000
```

1st Simulation

Let's titrate a protein with a ligand. The protein is around 20μl, and the stock solution of ligand is 1 mM.

we titrate the ligand from 0 to $\approx 100\mu$M

In the prpgram, CP CL are concentration of Prot and ligand. V is total volume. dV is aliquote volume.

all uncertainties are prefixed with a u.

Pipeting

assume 10 aliquotes of 10 μ l at 1 mM starting from 500 μ l

1) first aliquoting into one tube

```
# xP : protein xL : ligand  uX : uncertainty
CPi = 20.0               # initial conc of constant (Protein) species
into sol
Vi = 500.0               # initial volume
CLstock = 1000.0          # stock conc of added (Ligand) species
dV = 5.0                 # aliquote volume
#-------------
CP = CPi             # current conc of Prot
uCPi = uncert_conc(CPi,0,1,0)
uCP = uCPi
V = Vi                   # current volume
uVi = uncert_vol(Vi)
uV = uVi
udV = uncert_vol(dV)
uCLstock = uncert_conc(CLstock,0,1,0)
CL = 0.0            # current conc of varying species into sol
uCL = 0.0
for ali in range(11):
    # first print current state
    print "Aliquote",ali," Prot", ustr(CP, uCP), "  Ligand", ustr(CL,
uCL), "  Vol", ustr(V, uV,"ul")
    # addition of dV into V
    CP = cp(CP, 0.0, V, dV)
    uCP = uncert_conc(CP, 0.0, V, dV, uCP, 0.0, uV, udV, covc12=1.0)
    CL = cp(CL, CLstock, V, dV)
    uCL = uncert_conc(CL, CLstock, V, dV, uCL, uCLstock, uV, udV,
covc12=1.0)
    V = V + dV
    uV = uadd(uV, udV, cov=1.0) + trf_err*V
```

```
Aliquote 0  Prot 20.00 uM +/- 2.00   Ligand 0.00 uM +/- 0.00   Vol
500.00 ul +/- 4.88
Aliquote 1  Prot 19.80 uM +/- 1.98   Ligand 9.90 uM +/- 1.00   Vol
505.00 ul +/- 17.56
Aliquote 2  Prot 19.61 uM +/- 1.96   Ligand 19.61 uM +/- 2.00   Vol
510.00 ul +/- 30.38
Aliquote 3  Prot 19.42 uM +/- 1.94   Ligand 29.13 uM +/- 3.01   Vol
515.00 ul +/- 43.31
Aliquote 4  Prot 19.23 uM +/- 1.92   Ligand 38.46 uM +/- 4.02   Vol
520.00 ul +/- 56.38
Aliquote 5  Prot 19.05 uM +/- 1.90   Ligand 47.62 uM +/- 5.03   Vol
525.00 ul +/- 69.56
Aliquote 6  Prot 18.87 uM +/- 1.89   Ligand 56.60 uM +/- 6.04   Vol
530.00 ul +/- 82.88
Aliquote 7  Prot 18.69 uM +/- 1.87   Ligand 65.42 uM +/- 7.05   Vol
535.00 ul +/- 96.31
Aliquote 8  Prot 18.52 uM +/- 1.85   Ligand 74.07 uM +/- 8.06   Vol
```

```
    540.00 ul +/- 109.88
    Aliquote 9  Prot 18.35 uM +/- 1.84   Ligand 82.57 uM +/- 9.06   Vol
    545.00 ul +/- 123.56
    Aliquote 10  Prot 18.18 uM +/- 1.82   Ligand 90.91 uM +/- 10.06
    Vol 550.00 ul +/- 137.38
```

2) making 11 different tubes :

```
# xP : protein xL : ligand  uX : uncertainty
CPi = 20.0               # initial conc of constant (Protein) species
into sol
Vi = 500.0               # initial volume
CLstock = 1000.0          # stock conc of added (Ligand) species
dV = 5.0                 # aliquote volume
#-------------
CP = CPi             # current conc of Prot
uCPi = uncert_conc(CPi,0,1,0)

uCP = uCPi
V = Vi                   # current volume
uVi = uncert_vol(Vi)
uV = uVi
udV = uncert_vol(dV)
uCLstock = uncert_conc(CLstock,0,100,0)
CL = 0.0            # current conc of varying species into sol
uCL = 0.0
for ali in range(11):
    cdV = ali*dV     # what is added in this tube
    ucdV = uncert_vol(cdV)
    # addition of cdV into Vi
    V = Vi + cdV
    uV = uadd(uVi, ucdV)
    CP = cp(CPi, 0.0, Vi, cdV)
    uCP = uncert_conc(CPi, 0.0, Vi, cdV, uCPi, 0.0, uVi, ucdV)
    CL = cp(0.0, CLstock, Vi, cdV)
    uCL = uncert_conc(0.0, CLstock, Vi, cdV,   0.0, uCLstock, uVi, ucdV)

    print "Tube",ali," Prot", ustr(CP, uCP), "  Ligand", ustr(CL, uCL),
"  Vol", ustr(V, uV,"ul")
```

```
    Tube 0  Prot 20.00 uM +/- 2.00   Ligand 0.00 uM +/- 0.00   Vol
    500.00 ul +/- 4.88
    Tube 1  Prot 19.80 uM +/- 1.98   Ligand 9.90 uM +/- 1.00   Vol
    505.00 ul +/- 4.88
    Tube 2  Prot 19.61 uM +/- 1.96   Ligand 19.61 uM +/- 1.98   Vol
    510.00 ul +/- 4.88
    Tube 3  Prot 19.42 uM +/- 1.94   Ligand 29.13 uM +/- 2.94   Vol
    515.00 ul +/- 4.88
    Tube 4  Prot 19.23 uM +/- 1.92   Ligand 38.46 uM +/- 3.88   Vol
    520.00 ul +/- 4.88
    Tube 5  Prot 19.05 uM +/- 1.90   Ligand 47.62 uM +/- 4.83   Vol
    525.00 ul +/- 4.89
    Tube 6  Prot 18.87 uM +/- 1.89   Ligand 56.60 uM +/- 5.72   Vol
    530.00 ul +/- 4.89
    Tube 7  Prot 18.69 uM +/- 1.87   Ligand 65.42 uM +/- 6.60   Vol
    535.00 ul +/- 4.89
    Tube 8  Prot 18.52 uM +/- 1.85   Ligand 74.07 uM +/- 7.47   Vol
    540.00 ul +/- 4.89
    Tube 9  Prot 18.35 uM +/- 1.83   Ligand 82.57 uM +/- 8.32   Vol
    545.00 ul +/- 4.89
```

```
Tube 10  Prot 18.18 uM +/- 1.82   Ligand 90.91 uM +/- 9.15   Vol
550.00 ul +/- 4.89
```

As you see here, concentrations error bars are quite different in both approaches.

2nd simulation

Here we do the real case from the paper (Table 2)

capillary tubes are used, which can be used with 50 μl of solution; we'll prepare 75 μl so that we have enough to fill the tube.

There is one protein stock solution, and 3 peptide stock solutions at 3 different concentrations.

```
# define the concentration and error of the peptide in the stock solution
Lo = 4500
uLo = 4.5*Lo/100
p4500 = [Lo, uLo]    # the 3 stock solutions

# then lets prepare the 2 additional dilutions :
p450 = [cp(Lo, 0.0, 20.0, 180.0), uncert_conc(Lo, 0.0, 20.0, 180.0,
uc1=uLo)]
p45 = [cp(Lo, 0.0, 2.0, 198.0), uncert_conc(Lo, 0.0, 2.0, 198.0,
uc1=uLo)]

Stocks = [p4500, p450, p45]    # list of stocks will be used in one tube
simul
print "Ligand stock solutions :"
for i in range(3):
    print "    Stock %d : %s"%(i+1, ustr(*Stocks[i]))
```

```
Ligand stock solutions :
    Stock 1 : 4500.00 uM +/- 202.50
    Stock 2 : 450.00 uM +/- 20.90
    Stock 3 : 45.00 uM +/- 2.16
```

```
# prep is the array to store the 11 prepared tubes
#prep[] codes for receipe of each tube
#[ [CLstock,uCLstock], VProt Vlig Vbuffer ]

prep = range(12)
prep[1] = [p45, 15, 0, 60]
prep[2] = [p45, 15, 18, 42]
prep[3] = [p450, 15, 3, 57]
prep[4] = [p450, 15, 6, 54]
prep[5] = [p450, 15, 15, 45]
prep[6] = [p4500, 15, 3, 57]
prep[7] = [p4500, 15, 6, 54]
prep[8] = [p4500, 15, 12, 48]
prep[9] = [p4500, 15, 16, 44]
prep[10] = [p4500, 15, 30, 30]
prep[11] =[p4500, 15, 50, 10]
CPstock = 64.4*75/15
print "CPstock: %.1f uM"%CPstock
VProtTot = 0.0    # used to accumulate total volume of protein solution
MLigTot = 0.0    # used to accumulate total quantity of peptide
CLv1 = []        # used to accumulate results
```

```
for i in range(1,12):
    [[CLstock, uCLstock], VProt, Vlig, Vbuffer ] = prep[i]
    VProtTot += VProt
    MLigTot += CLstock*Vlig
    Ci = cp(CLstock, 0.0, Vlig, Vbuffer)
    uCi = uncert_conc(CLstock, 0.0, Vlig, Vbuffer, uc1=uCLstock)
    Vi = Vlig + Vbuffer
    uVi = uadd( uncert_vol(Vlig), uncert_vol(Vbuffer))
    CL = cp( Ci, 0.0, Vi, VProt)
    uCL = uncert_conc(Ci, 0.0, Vi, VProt, uc1=uCi, uv1=uVi)
    CLv1.append((CL,uCL))
    VT = Vi + VProt
    uVT = uadd( uVi, uncert_vol(VProt))
    CP = cp(CPstock, 0.0, VProt, Vi)
    uCP = uncert_conc(CPstock, 0.0, VProt, Vi, uv2=uVi)
    print "Tube %d : Prot %s  Peptide %s    Vol %s"% (i, ustr(CP,uCP),
ustr(CL,uCL), ustr(VT, uVT,"ul"))
print "Total volume of protein : %.1f ul at %.0f uM : %.1f nanomole."%
(VProtTot,CPstock,VProtTot*CPstock/1000)
print "Total amount of peptide : %.1f nanomole."%(MLigTot/1000)
```

```
CPstock: 322.0 uM
Tube 1 : Prot 64.40 uM +/- 6.47  Peptide 0.00 uM +/- 0.00    Vol
75.00 ul +/- 0.50
Tube 2 : Prot 64.40 uM +/- 6.47  Peptide 10.80 uM +/- 0.53    Vol
75.00 ul +/- 0.45
Tube 3 : Prot 64.40 uM +/- 6.47  Peptide 18.00 uM +/- 0.89    Vol
75.00 ul +/- 0.48
Tube 4 : Prot 64.40 uM +/- 6.47  Peptide 36.00 uM +/- 1.74    Vol
75.00 ul +/- 0.46
Tube 5 : Prot 64.40 uM +/- 6.47  Peptide 90.00 uM +/- 4.28    Vol
75.00 ul +/- 0.45
Tube 6 : Prot 64.40 uM +/- 6.47  Peptide 180.00 uM +/- 8.66    Vol
75.00 ul +/- 0.48
Tube 7 : Prot 64.40 uM +/- 6.47  Peptide 360.00 uM +/- 16.86    Vol
75.00 ul +/- 0.46
Tube 8 : Prot 64.40 uM +/- 6.47  Peptide 720.00 uM +/- 33.30    Vol
75.00 ul +/- 0.44
Tube 9 : Prot 64.40 uM +/- 6.47  Peptide 960.00 uM +/- 44.28    Vol
75.00 ul +/- 0.45
Tube 10 : Prot 64.40 uM +/- 6.48  Peptide 1800.00 uM +/- 82.67
Vol 75.00 ul +/- 0.55
Tube 11 : Prot 64.40 uM +/- 6.47  Peptide 3000.00 uM +/- 135.35
Vol 75.00 ul +/- 0.44
Total volume of protein : 165.0 ul at 322 uM : 53.1 nanomole.
Total amount of peptide : 538.1 nanomole.
```

Doing the same in one tube is another story.

Let's assume we have the same stock solutions, however 2 differences

- working in capilaries is not possible, we have to change to $5mm$ tubes which have to be filed at 550μl
- stock solution of peptide contain also the protein (at the same concentration as in tube)

Protocole is as follow : in the one tube, add a given volume v_S of the stock solution, so that the protein is at the same conc as in previous experiment

Each aliquote is computed so that the target peptides concentrations are correct.

- BIG LETTERS are for n tubes experiment, small letters are for one tube exp. n is for the n^{th} aliquote. m is $n-1$
- C_S is the stock concentration

- C_n is the target concentration
- c_m is the previous concentration
- v_m is the previous volume
- v_S is the volume of stock solution to add

We have first to compute v_S analytically :

```
var("C_n ,C_S, c_m, v_m, v_S")

c_n = cp(C_S,c_m,v_S,v_m)        # one tube

E = (C_n == c_n)
# which volume v_S to add so that concentration is equal to target ?
show(solve(E,v_S))
```

$$\left[v_S = \frac{(C_n - c_m)v_m}{C_S - C_n}\right]$$

```
VProtTot = 0.0
VLTot = 0.0
CLv2 = []

for i in range(1,12):
    # 1st tube is special
    if i == 1:
        pp = copy(prep[i])    # start with tube 1
        for v in (1,2,3):    # and multiply all the volumes
            pp[v] *= (550./75)
        [[CLstock, uCLstock], VProt, Vlig, Vbuffer ] = pp
        CL = 0.0
        uCL = 0.0
        VT = VProt + Vbuffer
        uVT = uadd( uncert_vol(VProt), uncert_vol(Vbuffer))
        CP = cp(CPstock, 0.0, VProt, Vbuffer)
        uCP = uncert_conc(CPstock, 0.0, VProt, Vbuffer)
        vs = 0.0
        VProtTot += VProt

    else:
        stock = 0    # take first stock solution
        [CLstock, uCLstock] = Stocks[stock]
        [[tCLstock, tuCLstock], tVProt, tVlig, tVbuffer ] = prep[i]   #
get value in temporary variables
        # compute target peptide conc :
        target = cp( tCLstock, 0.0, tVlig, tVProt+tVbuffer) # This is
the target concentration == C_n
        #compute new values
        vs = (target-CL)*VT/(CLstock-target)    # this is equation
above, gives volume to add.
        if vs<10 and CLstock>45:
            stock += 1    # take next stock solution
            [CLstock, uCLstock] = Stocks[stock]
            vs = (target-CL)*VT/(CLstock-target)    # this is equation
above, gives volume to add.

        Cn = cp( CLstock, CL, vs, VT)    # we add the aliquote
        uCL = uncert_conc(CLstock, CL, vs, VT, uc1=uCLstock, uc2=uCL,
```

```
uv2=uVT)
        CL = Cn

        VT = VT + vs
        uVT = uadd( uVT, uncert_vol(vs)) + trf_err*VT

        CP = cp(CP, CP, vs, VT)
        uCP = uncert_conc(CP, CP, vs, VT, covc12=1.0, uv2=uVT)

        VProtTot += vs*CP/CPstock
        VLTot += vs
    CLv2.append((CL,uCL))
    print "aliquote %d : added %.2f   Prot %s  Peptide %s    Vol %s"%
(i, vs, ustr(CP,uCP), ustr(CL,uCL), ustr(VT, uVT,"ul"))


print "Total volume of ligand : %.1f ul."%(VLTot)
VProtTot += VLTot*CP /CPstock
print "Total volume of protein : %.1f ul at %.0f uM : %.1f nanomole."%
(VProtTot,CPstock,VProtTot*CPstock/1000)
```

```
    aliquote 1 : added 0.00   Prot 64.40 uM +/- 6.47  Peptide 0.00 uM
    +/- 0.00    Vol 550.00 ul +/- 4.59
    aliquote 2 : added 13.52   Prot 64.40 uM +/- 6.44  Peptide 10.80 uM
    +/- 0.52    Vol 563.52 ul +/- 18.68
    aliquote 3 : added 9.39   Prot 64.40 uM +/- 6.44  Peptide 18.00 uM
    +/- 0.66    Vol 572.92 ul +/- 33.00
    aliquote 4 : added 24.91   Prot 64.40 uM +/- 6.44  Peptide 36.00 uM
    +/- 1.49    Vol 597.83 ul +/- 47.95
    aliquote 5 : added 89.67   Prot 64.40 uM +/- 6.44  Peptide 90.00 uM
    +/- 4.84    Vol 687.50 ul +/- 65.14
    aliquote 6 : added 14.32   Prot 64.40 uM +/- 6.44  Peptide 180.00 uM
    +/- 10.49    Vol 701.82 ul +/- 82.68
    aliquote 7 : added 30.51   Prot 64.40 uM +/- 6.44  Peptide 360.00 uM
    +/- 24.29    Vol 732.34 ul +/- 100.99
    aliquote 8 : added 69.75   Prot 64.40 uM +/- 6.44  Peptide 720.00 uM
    +/- 53.51    Vol 802.08 ul +/- 121.05
    aliquote 9 : added 54.38   Prot 64.40 uM +/- 6.44  Peptide 960.00 uM
    +/- 61.89    Vol 856.46 ul +/- 142.46
    aliquote 10 : added 266.45   Prot 64.40 uM +/- 6.44  Peptide 1800.00
    uM +/- 126.34    Vol 1122.92 ul +/- 170.57
    aliquote 11 : added 898.33   Prot 64.40 uM +/- 6.44  Peptide 3000.00
    uM +/- 152.68    Vol 2021.25 ul +/- 221.26
    Total volume of ligand : 1471.3 ul.
    Total volume of protein : 698.5 ul at 322 uM : 224.9 nanomole.
```

```
def printCLV(cv):
    errm = 0
    for i, (C,uC) in enumerate(cv):
        if C != 0:
            err = 100.0*uC/C
        else:
            err = 0
        errm += err
        print "tube %d  : %s  err : %.1f %%"%(i,ustr(C,uC),err)
    print "mean error : %.1f %%"%( errm/(len(cv)-1) )

print "n capillaries"
printCLV(CLv1)
print "\none tube"
printCLV(CLv2)
```

```
n capillaries
tube 0  : 0.00 uM +/- 0.00  err : 0.0 %
tube 1  : 10.80 uM +/- 0.53  err : 4.9 %
tube 2  : 18.00 uM +/- 0.89  err : 4.9 %
tube 3  : 36.00 uM +/- 1.74  err : 4.8 %
tube 4  : 90.00 uM +/- 4.28  err : 4.8 %
tube 5  : 180.00 uM +/- 8.66  err : 4.8 %
tube 6  : 360.00 uM +/- 16.86  err : 4.7 %
tube 7  : 720.00 uM +/- 33.30  err : 4.6 %
tube 8  : 960.00 uM +/- 44.28  err : 4.6 %
tube 9  : 1800.00 uM +/- 82.67  err : 4.6 %
tube 10  : 3000.00 uM +/- 135.35  err : 4.5 %
mean error : 4.7 %

one tube
tube 0  : 0.00 uM +/- 0.00  err : 0.0 %
tube 1  : 10.80 uM +/- 0.52  err : 4.8 %
tube 2  : 18.00 uM +/- 0.66  err : 3.7 %
tube 3  : 36.00 uM +/- 1.49  err : 4.1 %
tube 4  : 90.00 uM +/- 4.84  err : 5.4 %
tube 5  : 180.00 uM +/- 10.49  err : 5.8 %
tube 6  : 360.00 uM +/- 24.29  err : 6.7 %
tube 7  : 720.00 uM +/- 53.51  err : 7.4 %
tube 8  : 960.00 uM +/- 61.89  err : 6.4 %
tube 9  : 1800.00 uM +/- 126.34  err : 7.0 %
tube 10  : 3000.00 uM +/- 152.68  err : 5.1 %
mean error : 5.7 %
```