Chiffrement/déchiffrement par substitutiton de lettres au lycée

Auteur : Raymond Moché

L'objet de ce papier est de repérer dans le domaine du chiffrement et déchiffrement des messages, ce qui pourrait être développé en activité d'algorithmique pour le lycée, voir 7. C'est pour cette raison que nous nous limiterons *a priori* au chiffrement par substitution [1] de lettres [2], qui est le plus facile. C'est un procédé fort ancien qui n'est plus utilisé car les messages codés par substitution ont la réputation de pouvoir être cassés par analyse fréquentielle, voir [11] ou [14] par exemple. Nous verrons ce qu'il en est en réalité, voir 6.

La méthode des fréquences, voir 5.2.2 et 6, n'est pas modélisée. Elle n'a donc aucune justification mathématique, voir 7.2.

Le logiciel de calcul utilisé ici est *scilab* [9] [3].

Commandes *scilab* **utilisées** : commandes basiques, boucles `pour` et `tant que`, `factorial`, `asciimat`, `taille`, `find`, `part`, appel d'une fonction *scilab*, concaténation de deux chaînes de caractères.

Mots-clefs : substitution, permutation, effectif, fréquence.

Table des matières

1. ou permutation
2. On dira simplement codage par substitution.
3. En général, on passe facilement d'un logiciel de calcul à un autre.

1 Position du problème

Voici un texte de Jules César [6] :

> Quand ces embarcations furent prêtes, César les fit transporter sur des chariots couplés à une distance de vingt-deux milles de son camp. Il les utilisa pour transporter sur l'autre rive un détachement qui s'empara par surprise d'une hauteur attenante au fleuve. Il se hâte de fortifier la position avant que l'ennemi s'aperçoive de rien, y fait passer une légion et, poussant les travaux des deux côtés à la fois, il rétablit le pont en deux jours. Par ce moyen il permit aux renforts et à ses fourrageurs de le rejoindre en toute sécurité et le ravitaillement des troupes devint plus facile

Imaginons qu'il ait voulu le transmettre à un destinataire qui devrait être capable de le comprendre sans qu'une tierce personne ne le puisse (en cas d'interception par exemple). Pour cela, le message sera codé par César et décodé par son destinataire, qui devra connaître d'avance la clef du décodage.

Pratiquement, on néglige les accents, les cédilles, les blancs et la ponctuation si bien que c'est en réalité ce texte

Listing 1 – Message de César

```
M=
quandcesembarcationsfurentpretescesarlesfittransportersurdescharioscoupl
esaunedistancedevingtdeuxmillesdesoncampillesutilisapourtransportersurlau
trerivundetachementquisemparaparsurprisedunehauteurattenanteaufleuveilse
hatadefortifierlapositionavantquelennemisapercoivederienyfaitpasseruneleg
ionetpoussantlestravauxdesdeuxcotesalafoisilretablitlepontendeuxjoursparc
emoyenilpermitauxrenfortsetasesfourrageursdelerejoindreentoutesecuriteetl
eravitaillementdestroupesdevintplusfacile
-->
```

que l'on devra coder.

2 Principe du codage par substitution de lettres

2.1 Généralités

Le *chiffrement*[4] *par substitution* [14] est un procédé de codage très simple consiste à effectuer une permutation convenue à l'avance[5] des lettres de l'alphabet. Il y a évidemment 26 ! chiffrements de ce type, soit :

4. Codage et chiffrement sont des mots synonymes.
5. avec le destinataire

```
-->factorial(26)
 ans  =
    4.032914611D+26
-->
```

2.2 Clef d'un codage par substitution

Pour décrire cette permutation, il suffit d'écrire en ligne les 26 lettres de l'alphabet et, en-dessous de chacune d'elle, la lettre qui lui correspond dans la permutation. Cette deuxième ligne peut être appelée *clef du chiffrement*[6].

Exemple fantaisiste : *Clef du codage* `azerty`[7] :

Listing 2 – Clef du codage `azerty`

```
abcdefghijklmnopqrstuvwxyz
azertyuiopqsdfghjklmwxcvbn
```

Nous verrons plus loin d'autres exemples de clefs de codage par substitution.

2.3 Clef d'un codage par décalage

2.3.1 Généralités

Le codage par décalage, (voir [13]), est un cas particulier de codage par substitution. Plus précisément, `j` étant un entier donné entre 1 et 25, la première lettre de la clef du chiffrement par décalage de pas `j` est la lettre de l'alphabet obtenue en décalant la lettre `a` de `j` pas vers la droite dans l'alphabet, la deuxième est la lettre de l'alphabet obtenue en décalant la lettre `b` de `j` pas vers la droite, *etc*, jusqu'à `z`[8].

2.3.2 Exemple : chiffrement de César

C'est le chiffrement par décalage de pas 3, voir [13], dont voici la clef :

Listing 3 – Clef du chiffrement de César

```
abcdefghijklmnopqrstuvwxyz
defghijklmnopqrstuvwxyzabc
```

3 Codage d'un message par décalage

Le codage se fait à la main si le message est court. Sinon un petit script *scilab* règlera le problème.

3.1 Codage de César à la main

Soit à chiffrer

nostradamus

6. Quelquefois, nous appellerons clef l'ensemble des 2 lignes.
7. Clef faisant référence aux claviers français.
8. On a besoin de supposer, à un certain moment, que `abcdefghijklmnopqrstuvwxyz` a été prolongé en `abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz`.

Il suffit de se reporter à la clef du codage de César ci-dessus : la première lettre `n` du message (à repérer sur la première ligne du listing 3) doit être remplacée par la lettre qui se trouve juste en dessous, soit `q`, la seconde, `o`, doit de même être remplacée par `r` et ainsi de suite, ce qui donne le message codé suivant :

qrvwudgdpxv

3.2 Codage de César avec *scilab*

Nous allons coder le message du listing 1 à l'aide du chiffrement de César. Ce texte étant assez long, nous programmerons le codage au lieu de l'effectuer à la main et obtiendrons une fonction *scilab* que nous appellerons `CoDec`[9] qui servira en fait pour tous nos codages et décodages.

3.3 La fonction `CoDec`

Listing 4 – Fonction `CoDec`

```
function MC=CoDec(M,A,C)
a=asciimat(A);
c=asciimat(C);
m=asciimat(M);
t=taille(m);
mC=[];
for  j=1:t
     x=c(find(a==m(j)));
     mC=[mC,x];
end
MC=asciimat(mC);
endfunction
```

Commentaires sur le script de `CoDec` *:*

Ligne 1 : La fonction `CoDec` a pour entrées 3 chaînes de caractères `M`, `A` et `C`. `M` est le message à coder ; `A` et `C` n'ont pas de doublons et ont la même longueur. `A` comprend tous les caractères utilisés dans `M`. Dans ce papier, au début, `A` est l'alphabet, en lettres minuscules, dans l'ordre lexicographique, `C` est la clef du codage considéré. `CoDec` retourne le message chiffré `MC`.
Ligne 2, 3 et 4 : La commande `asciimat` produit, à partir d'une chaîne de caractères, le vecteur des codes `ASCII`[10] de ces caractères, et inversement[11]. Ce détour[12] permet de remplacer un travail sur des chaînes de caractères par un travail sur des vecteurs de nombres entiers.
Ligne 5 : Longueur du message à coder.
Ligne 6 : Initialisation du vecteur `mC` des codes `ASCII` des caractères du message codé.
Ligne 7 à 10 : Pour chaque caractère du message, sous sa forme de code `ASCII`, on calcule son rang dans `A` et on identifie le caractère de la clef de même rang (celui qui se trouve juste en-dessous), qui est un nombre entier compris entre 97 et 122 ; enfin, on ajoute ce nombre à la fin de `mC`.
Ligne 11 : Finalement, on met le message codé sous la forme d'une chaîne de caractères notée `MC` en remplaçant les codes ASCII composant `mC` par les caractères correspondants.

9. pour codage-décodage
10. Voir [12]. Les lettres accentuées n'ont pas de code `ASCII` ainsi que certains signes.
11. Par exemple, si `A='abcdefghijklmnopqrstuvwxyz'`, `asciimat(A)` est le vecteur `[97,...,122]` tandis que `asciimat([97, ... ,122])=A`.
12. Sauf erreur, il est impossible de faire ce travail directement sur des chaînes de caractères avec *scilab* parce que l'on manque d'une commande analogue à `find`.

Listing 5 – Script du codage du message de César

```
M='quandcesembarcationsfurentpretescesarlesfittransportersurdescharIotsco
uplesaunedistancedevingtdeuxmillesdesoncampillesutilisapourtransportersur
lautreriveundetachementquisemparaparsurprisedunehauteurattenanreaufleuvei
lsehatadefortifierlapositionavantquelennemisapercoivederienyfaitpasserune
legionetpoussantlestravauxdesdeuxcotesalafoisilretablitlepontendeuxjoursp
arcemoyenilpermitauxrenfortsetasesfourrageursdelerejoindreentoutesecurite
etleravitaillementdestroupesdevintplusfacile';
exec('Chemin de la fonction CoDec.sci', -1);
A='abcdefghijklmnopqrstuvwxyz';
C='defghijklmnopqrstuvwxyzabc';
MC=CoDec(M,A,C);
afficher('MC=');
afficher(MC);
```

On remarquera ce script comprend le chargement de la fonction `CoDec`. Voici ce qu'il retourne :

Listing 6 – Message codé de César

```
-->exec('Chemin du script ci-dessus', -1)
 MC=
 txdqgfhvhpedufdwlrqvixuhqwsuhwhvfhvduohvilwwudqvsruwhuvxughvfkdulrwvfrxs
 ohvdxqhglvwdqfhghylqjwghxaploohvghvrqfdpsloohvxwlolvdsrxuwudqvsruwhuvxuo
 dxwuhulyhxqghwdfkhphqwtxlvhpsdudsduvxusulvhgxqhkdxwhxudwwhqdquhdxiohxyhl
 ovhkdwdghiruwlilhuodsrvlwlrqdydqwtxhohqqhplvdshufrlyhghulhqbidlwsdvvhuxq
 hohjlrqhwsrxvvdqwohvwudydxaghvghxafrwhvdodirlvlouhwdeolwohsrqwhqghxamrxu
 vsdufhprbhqloshuplwdxauhqiruwvhwdvhvirxuudjhxuvghohuhmrlqguhhqwrxwhvhfxu
 lwhhwohudylwdloohphqwghvwurxshvghylqwsoxvidfloh
-->
```

Il n'y a rien de neuf.

4 Codage par substitution de lettres quelconque

Se reporter à la section précédente. Il suffit d'utiliser la clef *ad hoc*. On peut utiliser comme clef toute chaîne de 26 signes ayant un code ASCII. Voici un exemple de codage exotique tapé directement dans la console :

Listing 7 – Codage exotique

```
-->A='abcdefghijklmnopqrstuvwxyz';
-->Clefexotique='!#$%&*+,-./:;<=>,@\^A&b2E3)';
-->M='leshabitsducardinal';
-->exec('Chemin de la fonction CoDec.sci', -1)
-->MC=CoDec(M,A, Clefexotique);
-->afficher('MC=');
 MC=
-->afficher(MC);
  :&\,!#-^\%A$!@%-<!:
-->
```

Ce codage n'est pas plus difficile à casser qu'un codage par substitution ordinaire.

5 Décodage d'un message par décalage

5.1 Décodage d'un décalage à pas connu

5.1.1 Décodage à la main

Reprenons par exemple le message codé

qrvwudgdpxv

On sait qu'il a été codé à l'aide du chiffrement de César. On se reporte donc au listing 3 : la première lettre, soit `q`, du message codé (à repérer sur la deuxième ligne de la clef) doit être remplacée par la lettre qui se trouve juste au-dessus, soit `n`, parce que `n` a donné `q` dans le codage ; la seconde, soit `r`, doit de même être remplacée par la lettre qui se trouve juste au-dessus, soit `o` et ainsi de suite. On obtient ainsi le message décodé, à savoir :

nostradamus

La méthode à la main est longue et ne peut convenir qu'à des messages très courts.

5.1.2 La fonction `CoDec` est une fonction de codage-décodage :

La remarque précédente vaut pour décoder n'importe quel message qui a été codé par substitution de lettres, quelle que soit cette substitution. Par conséquent, en toute généralité, si `MC=CoDec(M,A,C)`, `M=CoDec(MC,C,A)`. `CoDec` est donc une fonction de codage-décodage.

5.1.3 Décodage à l'aide de *scilab*

À titre d'exemple, décodons le message codé de César (cf. listing 6) à l'aide du script suivant, qui a été rédigé dans l'éditeur de texte *SciNotes* à la suite du listing 5 :

Listing 8 – suite du listing 5

```
X=CoDec(MC,C,A);
afficher('X=');
afficher(X);
```

On obtient :

```
X=
   quandcesembarcationsfurentpretescesarlesfittransportersurdeschariot
scouplesaunedistancedevingtdeuxmillesdesoncampillesutilisapourtransporter
surlautreriveundetachementquisemparaparsurprisedunehauteurattenanteaufleu
veilsehatedefortifierlapositionavantquelennemisapercoivederienyfaitpasser
unelegionetpoussantlestravauxdesdeuxcotesalafoisilretablitleponten deuxjou
rsparcemoyenilpermitauxrenfortsetasesfourrageursdelerejoindreentoutesecur
iteetleravitaillementdestroupesdevintplusfacile
-->
```

On constate que `X=M` : on a retrouvé le message initial.

5.2 Décodage d'un décalage à pas inconnu

Essayons de déchiffrer le message suivant :

Listing 9 – Un message codé par décalage de pas inconnu

```
MC='fytfdszveuvdrizejtfdszveuvtrgzkrzevjhlzjfekgrikzjafpvlogfliuvjtflijvj
cfzekrzevj'
```

sachant seulement qu'il a été codé par décalage[13].

5.2.1 Décodage par essai de tous les chiffrements par décalage, avec *scilab*

Comme il n'y a que 25 chiffrements par décalage, on peut les essayer tous à l'aide d'une boucle `pour`. On peut donc utiliser le script suivant :

Listing 10 – Essai de tous les décalages

```
A='abcdefghijklmnopqrstuvwxyz';
ADouble=A+A;
MC='fytfdszveuvdrizejtfdszveuvtrgzkrzevjhlzjfekgrikzjafpvlogfliuvjtflijvj
cfzekrzevj';
for j=1:25
    C=part(ADouble,j+1:j+26);
    X=CoDec(MC,C,A);
    afficher(X);
end
```

[14]. Ensuite, on charge la fonction `CoDec` et on applique le script, ce qui donne :

```
-->exec('Chemin de CoDec.sci', -1)
-->exec('Chemin du script', -1)
 exsecryudtucqhydisecryudtusqfyjqyduigkyiedjfqhjyizeouknfekhtuisekhiuibey
 djqydui
 dwrdbqxtcstbpgxchrdbqxtcstrpexipxcthfjxhdciepgixhydntjmedjgsthrdjghthadx
 cipxcth
 cvqcapwsbrsaofwbgqcapwsbrsqodwhowbsgeiwgcbhdofhwgxcmsildcifrsgqcifgsgzcw
 bhowbsg
 bupbzovraqrznevafpbzovraqrpncvgnvarfdhvfbagcnegvfwblrhkcbheqrfpbhefrfybv
 agnvarf
 atoaynuqzpqymduzeoaynuqzpqombufmuzqecgueazfbmdfuevakqgjbagdpqeoagdeqexau
 zfmuzqe
 zsnzxmtpyopxlctydnzxmtpyopnlateltypdbftdzyealcetduzjpfiazfcopdnzfcdpdwzt
 yeltypd
 yrmywlsoxnowkbsxcmywlsoxnomkzsdksxocaescyxdzkbdsctyioehzyebnocmyebcocvys
 xdksxoc
 xqlxvkrnwmnvjarwblxvkrnwmnljyrcjrwnbzdrbxwcyjacrbsxhndgyxdamnblxdabnbuxr
 wcjrwnb
 wpkwujqmvlmuizqvakwujqmvlmkixqbiqvmaycqawvbxizbqarwgmcfxwczlmakwczamatwq
 vbiqvma
 vojvtiplukl thypuzjvtiplukljhwpahpulzxbpzvuawhyapzqvflbewvbyklzjvbyzlzsvp
 uahpulz
```

13. de pas inconnu
14. Commentaires sur le script « Essai de tous les décalages » :

• à la ligne 2, le signe `+` est le signe de concaténation des chaînes. Par conséquent, `A+A` est la chaîne `'abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz'` ;

• à la ligne 5, on extrait de la chaîne précédente la chaîne de ses éléments du rang `j+1` au rang `j+26` à l'aide de la commande `part`.

```
uniushoktjksgxotyiushoktjkigvozgotkywaoyutzvgxzoypuekadvuaxjkyiuaxykyruo
tzgotky
tmhtrgnjsijrfwnsxhtrgnjsijhfunyfnsjxvznxtsyufwynxotdjzcutzwijxhtzwxjxqtn
syfnsjx
slgsqfmirhiqevmrwgsqfmirhigetmxemriwuymwsrxtevxmwnsciybtsyvhiwgsyvwiwpsm
rxemriw
rkfrpelhqghpdulqvfrpelhqghfdslwdlqhvtxlvrqwsduwlvmrbhxasrxughvfrxuvhvorl
qwdlqhv
qjeqodkgpfgoctkpueqodkgpfgecrkvckpguswkuqpvrctvkulqagwzrqwtfgueqwtugunqk
pvckpgu
pidpncjfoefnbsjotdpncjfoefdbqjubjoftrvjtpouqbsujtkpzfvyqpvseftdpvstftmpj
oubjoft
ohcombiendemarinscombiendecapitainesquisontpartisjoyeuxpourdescourseslоi
ntaines
ngbnlahdmcdlzqhmrbnlahdmcdbzohszhmdrpthrnmsozqshrinxdtwontqcdrbntqrdrknh
mszhmdr
mfamkzgclbckypglqamkzgclbcayngryglcqosgqmlrnyprgqhmwcsvnmspbcqamspqcqjm
glryglcq
lezljyfbkabjxofkpzljyfbkabzxmfqxfkbpnrfplkqmxoqfpglvbrumlroabpzlropbpilf
kqxfkbp
kdykixeajzaiwnejoykixeajzaywlepwejaomqeokjplwnpeofkuaqtlkqnzaoykqnoaohke
jpwejao
jcxjhwdziyzhvmdinxjhwdziyzxvkdovdiznlpdnjiokvmodnejtzpskjpmyznxjpmnzngjd
iovdizn
ibwigvcyhxygulchmwigvcyhxywujcnuchymkocmihnjulncmdisyorjiolxymwiolmymfich
nuchym
havhfubxgwxftkbglvhfubxgwxvtibmtbgxljnblhgmitkmblchrxnqihnkwxlvhnklxlehb
gmtbgxl
gzugetawfvwesjafkugetawfvwushalsafwkimakgflhsjlakbgqwmphgmjvwkugmjkwkdg
aflsafwk
—>
```

On constate que seule la 17ème sortie est compréhensible. C'est donc le message original. Remarquons que *scilab* seul n'est pas capable de retourner le message décodé. Il faut comprendre le français. Il y a intervention humaine. Le message décodé, soit :

ohcombiendemarinscombiendecapitainesquisontpartisjoyeuxpourdescourseslointaines

peut maintenant être écrit en clair, à la main :

Oh ! Combien de marins, combien de capitaines qui sont partis joyeux pour des courses lointaines

On en déduit aussi que le pas du décalage qui a servi au chiffrement du message initial était 17.

5.2.2 Décodage par la méthode des fréquences, avec *scilab*

Fréquence des lettres en français, voir ([11]) [15]

15. On trouve ces fréquences sur beaucoup de sites qui, souvent, n'indiquent pas comment elles ont été calculées. Il y a des exceptions : le site de Bibm@th.net donne des indications assez précises, voir [5]. Le concept de fréquence d'une lettre dans une langue donnée reste quand même assez scabreux.

E	17,76	M	2,72
S	8,23	Q	1,34
A	7,68	V	1,27
N	7,61	G	1,10
T	7,30	F	1,06
I	7,23	B	0,80
R	6,81	H	0,64
U	6,05	X	0,54
L	5,89	Y	0,21
O	5,34	J	0,19
D	3,60	Z	0,07
C	3,32	K	0,00
P	3,24	W	0,00

Ce tableau appelle les remarques suivantes :

• La fréquence du `e`[16] est beaucoup plus grande que celle des autres lettres (0.1776). On peut donc espérer, *si l'on a à déchiffrer un message codé par substitution inconnue*[17], que la lettre la plus fréquente du message correspond à la lettre `e` (si l'on tire une lettre au hasard dans un long texte français standard, il y a un peu plus d'une chance sur 6 que ce soit un `e`).

• Cette remarque ne vaut évidemment que si les fréquences des lettres du message codé que l'on a calculées sont significatives (?), c'est à dire (?) si ce message est suffisamment long[18].

• Ces remarques de bon sens valent aussi pour les lettres suivantes plus difficiles à détecter.

• Les totaux des pourcentages des 2[19], 5[20] et 10[21] lettres les plus fréquentes de la langue française sont respectivement égaux à 25,99%, 48,52% et 79,89%, ce qui veut dire que si l'on tire une lettre au hasard dans un long texte en français, il y a à peu près une chance sur 4, une chance sur 2 et 4 chances sur 5 de tirer un `e`, un `e` ou un `s`, une lettre de `e,s,a,n,t` ou une lettre de `e,s,a,n,t,i,r,u,l,o`.

Exemple qui marche On se propose de déchiffrer le message suivant, codé par décalage de pas inconnu :

```
M='tsutzafnyjshtwjwjrfwvzjwtzywjhjyyjrfxxjifwwnafsyxvznatzqfnjsyjsywjwifs
xqfanqqjvzjqvzjxlwtzujxijufwynhzqnjwxvznxjrgqfnjsyjsjywjxtwynx';
```

Pour cela, nous allons rechercher la lettre de ce message dont la fréquence est la plus élevée. En fait, on va utiliser des effectifs au lieu de fréquences. On calculera les effectifs des différentes lettres du message à l'aide de la fonction `Effectifs` suivante :

Listing 11 – Effectifs

```
function lEf=Effectifs(M)
    m=asciimat(M);
    lEf=[];
    while taille(m)>0
        d=m($);
        x=find(m==d);
        tx=taille(x);
        lEf=[lEf,[d;tx]];
        m(x)=[];
    end
endfunction
```

Commentaires sur le script « Effectifs » :

• `M` est une chaîne de lettres.

16. comprenant les `e` accentués
17. en particulier par décalage de pas inconnu
18. Cette remarque communément admise n'a aucun fondement mathématique.
19. `e,s`
20. `e,s,a,n,t`
21. `e,s,a,n,t,i,r,u,l,o`

- `m` est la chaîne des codes `ASCII` de ces lettres.
- `lEf` initialise le tableau des différentes lettres de `M` (1ère ligne) et de leurs effectifs (2ème ligne).
- `d` est le dernier élément de `m`.
- `x` est le vecteur des rangs dans `m` où l'on trouve `x`.
- `tx` est l'effectif de `d` dans `m`.
- La commande `m(x)=[]` efface la lettre que l'on vient de traiter.

Ce qui donne :

```
-->clear
-->exec('Chemin de CoDec.sci', -1)
 -->M='tsutzafnyjshtwjwjrfwvzjwtzywjhjyyjrfxxjifwwnafsyxvznatzqfnjsyjsywj
wifsxqfanqqjvzjqvzjxlwtzujxijufwynhzqnjwxvznxjrgqfnjsyjsjywjxtwynx';
-->exec('Chemin de Effectifs.sci', -1)
-->lEf=Effectifs(M);afficher(lEf);
          column  1 to 12
    120.    110.    121.    119.    116.    106.    115.    102.    113.
     10.     10.     11.     14.      7.     22.      8.     10.      7.

          column 13 to 18
    103.    114.    122.    118.    104.    117.    105.    108.     97.
      1.      3.     10.      5.      3.      3.      3.      1.      4.
-->
```

On constate qu'il y a 18 lettres différentes dans le message `M` et que l'une d'elles a un effectif nettement plus grand que les 17 autres. C'est la lettre de code `ASCII` égal à 106, soit

```
-->ascii(106)
 ans  =
 j
-->
```

Cela signifie que dans le décalage du codage inconnu, la lettre `e` a été remplacée par `j`[22], autrement dit, le pas de ce décalage est 5. Admettant cette hypothèse, on peut décoder le message `M` comme précédemment. Si l'on obtient un texte en compréhensible en français, on conclura que c'est bien le message recherché[23].

```
-->clear
-->M='tsutzafnyjshtwjwjrfwvzjwtzywjhjyyjrfxxjifwwnafsyxvznatzqfnjsyjsywjw
ifsxqfanqqjvzjqzjxlwtzujxijufwynhzqnjwxvznxjrgqfnjsyjsjywjxtwynx';
-->exec('Chemin de Effectifs.sci', -1)
-->lEf=Effectifs(M);afficher(lEf);
          column  1 to 12
    120.    110.    121.    119.    116.    106.    115.    102.    113.
     10.     10.     11.     14.      7.     22.      8.     10.      7.

          column 13 to 18
    103.    114.    122.    118.    104.    117.    105.    108.     97.
      1.      3.     10.      5.      3.      3.      3.      1.      4.
-->A='abcdefghijklmnopqrstuvwxyz';
```

22. Ce n'est qu'une hypothèse, assez vraisemblable. Ce n'est pas une certitude.
23. en espérant qu'il n'y a qu'un seul message compréhensible !

```
-->C='fghijklmnopqrstuvwxyzabcde';
-->exec('Chemin de CoDec.sci', -1)
-->Message=CoDec(M,C,A);afficher(Message);
onpouvaitencoreremarqueroutrecettemassedarrivantsquivoulaiententrerdans
lavillequelquesgroupesdeparticuliersquisemblaientenetresortis
-->
```

Commentaires :

`Message` est compréhensible donc `C` est bien la clef du chiffrement par décalage utilisé.

Exemples qui ne marchent pas Tout message dont la fréquence de la lettre qui code le `e` n'est pas la plus élevée conduira à une erreur par la méthode précédente. On conclura quand même qu'en gros, c'est une méthode efficace.

6 Décodage par substitution de lettres

Cette section repose uniquement sur la méthode des fréquences.

6.1 Exemple

On désire maintenant déchiffrer le message codé `M0` suivant :

```
ostlmsthswlptwfttwkghttfatmktktmtfwhgwkwftdollogflhamoastetlmwfmgweitamgwm
hwoljwostlmetofmwktfgoktrtpwrgtmpgwtrwlavghigftrtlyaewsmtlrgfmosftrtxkaomhala
xgokztlgofrwdgoflhalawmafmjwtladaomkoltrtlovsafuwtlwfamgwmhgwkegddwfojwtka
xteltlegsstuwtlkwlltlgwadtkoeaoflosawkartjwgolgeewhtkhwoljwtlgfhkgukaddtrtmkaxa
osegdhmazosoltrtfgdzktwltltvhtkotfetlaoflostptwftalmkgfawmtltkaegfftemtawftmazst
mmthgwkegsstemtkrtlrgfftlktsamoxtlasahiblogsguotiwdaoftetltmwrtlltkxokgfmfgma
ddtfmartyofokrtyaegfmktlhkteolttmasgfumtkdtstlztlgoflasodtfmaoktlrtlalmkgfawmtls
gklrtlxgslrtsgfuwtrwkttmtsetswojwohgwkkapgofrktdaklwftawmktdollogfegflolmtkaam
tlmtksamitgkotrtsaukaxomamogfltsgfsajwtsstsaukaxomtawftofyswtfetlwkstegwstdtfm
rwmtdhlhgwkdtlwktkstmtdhlaxtewfttvaemomwrttmwftlmazosomtoftuasttwftigksguta
mgdojwtfgwxtsstltkaoflmassttastvmtkotwkrtsalmamogfgkzomastetmmttvhtkotfethtk
dtmmkartlbfeikgfoltkaxtewfthswlukafrthkteologfstllblmtdtlrthglomogfftdtfmhaklamts
somt
```

On sait seulement qu'il a été codé par une substitution dont la clef est inconnue. Ce problème est beaucoup plus difficile que le cas particulier d'un décalage de pas inconnu. En effet, les clefs de substitution sont tellement nombreuses que l'on ne peut pas envisager de les essayer toutes, comme on l'a fait en 5.2.2. Reste la méthode de décryptage basée sur les fréquences des lettres du message codé. Cette méthode des fréquences est réputée assez facile (cf. [14] par exemple). En fait, ce n'est pas vrai. Ça peut être très long. C'est aussi très désagréable, comme on va le voir dans l'exemple entièrement traité ci-dessous. Nous allons successivement :

- calculer la fréquence des différents caractères de `M0`,
- ranger ces caractères par ordre de fréquences décroissantes, ce qui conduira à `L1` puis à `Cideal` ci-dessous [24],
- remplacer dans `M0` chaque caractère de `Cideal` par le caractère de même rang de la chaîne

`Aideal='esantirulodcpmqvgfbhxyjzkw'`.

24. `Cideal` est la chaîne de toutes les lettres minuscules de l'alphabet rangées dans l'ordre de leurs fréquences dans `M0` décroissantes, les lettres absentes étant considérées comme présentes avec la fréquence nulle.

[25]. Identifier ainsi chaque caractère de `MO` avec le caractère de même rang [26] de `Aideal` fournirait, si cela marchait, une méthode de décryptage automatique (entièrement calculable sans intervention du déchiffreur). Mais cela ne marche pas comme on va le voir. Nous devrons décoder les lettres presque une par une !

Ces scripts successifs [27] sont des blocs de commandes rédigés au fur et à mesure dans *Scinotes* où ils se suivent à la queue leu leu. Le script complet s'appelle `Commandes.sce`. Il utilise les fonctions `CoDec` et `Resumer` qui doivent être chargées préalablement dans la console de *Scilab*. La fonction `Resumer` prolonge la fonction `Effectifs` en rangeant les lettres du message considéré par ordre de fréquences décroissantes [28] :

Listing 12 – Fonction `Resumer`

```
//Ch est une chaine de lettres donnee. La fonction renvoie le vecteur des
//codes ASCII des lettres figurant dans Ch rangees dans l'ordre des
//frequences decroissantes (ligne1) et ces frequences (ligne2).
function res=Resumer(Ch)
    ch=asciimat(Ch);//Vecteur des codes ASCII des lettres de Ch.
    t=taille(ch);
    Resu=[];//Initialisation de la matrice dont la premiere ligne sera le
    //vecteur des elements differents de ch (comptes a partir du dernier)
    //et dont la seconde sera la suite de leurs effectifs.
    chp=ch;
    while taille(chp)>0
        der=chp($);//Code ASCII de la derniere lettre de Ch.
        x=find(chp==der);//Rangs des apparitions de der dans chp.
        ef=taille(x);
        Resu=[Resu,[der;ef]];
        chp(x)=[];//Effacement de ce que l'on vient de traiter.
    end
    //On va maintenant ranger les codes des lettres figurant dans Ch par
    //ordre de frequences decroissantes.
    res=[];//Initialisation du vecteur des codes des lettres de Ch
    //rangees dans l'ordre decroissant.
    l1=Resu(1,:);//Codes ASCII des lettres figurant dans Ch, rangees.
    l2=Resu(2,:);//Effectifs de ces lettres.
    format(6);
    while taille(l2)>0
        m=max(l2);
        k=max(find(l2==m));//On choisit une lettre d'effectif maximum
        //s'il y en a plusieurs.
        res=[res,[l1(k);l2(k)/t]];
        l1(k)=[];
        l2(k)=[];
    end
endfunction
```

Nous pouvons commencer à dérouler le script `Commandes.sce` qui fournira le décodage recherché :

25. chaîne des lettres minuscules de l'alphabet rangées par ordre de fréquences décroissantes, voir [11]
26. rang lu dans `Cideal`
27. suivis de commentaires
28. Comme on le constate souvent, ce script commenté est plus indigeste que si l'on avait omis les commentaires.

Listing 13 – Début de `Commandes.sce`

```
clear
Alphabet='abcdefghijklmnopqrstuvwxyz';
M0='ostlmsthswlptwfttwkghttfatmktktmtfwhgwkwftdollogflhamoas..
tetlmwfmgweitamgwmhwoljwostlmetofmwktfgoktrtpwrgtmpgwtrwlavg..
higftrtlyaewsmtlrgfmosftrtxkaomhalaxgokztlgofrwdgoflhalawmaf..
mjwtladaomkoltrtlovsafuwtlwfamgwmhgwkegddwfojwtkaxteltlegsstu..
wtlkwlltlgwadtkoeaoflosawkartjwgolgeewhtkhwoljwtlgfhkgukaddtrt..
mkaxaosegdhmazosoltrtfgdzktwltltvhtkotfetlaoflostptwftalmkgfa..
wmtltkaegfftemtawftmazstmmthgwkegsstemtkrtlrgffttlktsamoxtlasa..
hiblogsguotiwdaoftetltmwrtlltkxokgfmfgmaddtfmartyofokrtyaegfmk..
tlhkteolttmasgfumtkdtstlztlgoflasodtfmaoktlrtlalmkgfawmtlsgklr..
tlxgslrtsgfuwtrwkttmtsetswojwohgwkkapgofrktdaklwftawmktdollogf..
egflolmtkaamtlmtksamitgkotrtsaukaxomamogfltsgfsajwtsstsaukaxom..
tawftofyswtfetlwkstegwstdtfmrwmtdhlhgwkdtlwktkstmtdhlaxtewfttv..
aemomwrttmwftlmazosomtoftuasttwftigksgutamgdojwtfgwxtsstltkaof..
lmassttastvmtkotwkrtsalmamogfgkzomastetmmttvhtkotfethtkdtmmkar..
tlbfeikgfoltkaxtewfthswlukafrthkteologfstllblmtdtlrthglomogfft..
dtfmhaklamtssomt';
exec('Chemin de la fonction Resumer.sci', -1)
r1=Resumer(M0);
//afficher('r1=');//affichage inutile.
//afficher(r1);//idem
l1=r1(1,:);//Vecteur des codes ASCII des lettres de M0 rangees par
//ordre de frequences decroissantes.
L1=asciimat(l1);
afficher('L1=');
afficher(L1);
f1=r1(2,:);//Vecteur des frequences correspondantes.
afficher('f1=');
afficher(f1);
```

ce qui donne :

Listing 14 – Résumé de `M0`

```
-->clear
-->exec('Chemin de Commandes.sce', -1)
 L1=
 tlamfowkgserhdxujzivpyb
 f1=
         column 1 to 10
    0.178    0.086    0.077    0.076    0.071    0.071    0.070    0.066
0.064    0.057
         column 11 to 20
    0.032    0.030    0.029    0.027    0.013    0.012    0.009    0.007
0.007    0.006
         column 21 to 23
    0.005    0.004    0.003
-->
```

Ces résultats montrent que le message codé `M0` utilise seulement 23 lettres de l'alphabet dont le rangement par ordre de fréquences décroissantes se lit sur la première ligne `l1` de `r1`, si on se

contente des codes ASCII de ces lettres. Elles-mêmes forment la chaîne `L1`, plus lisible.

Ci-dessous, le déchiffreur a complété `L1` à la main avec les 3 lettres manquantes `c,n,q`, ce qui a donné la chaîne `Cideal`[29] appelée ainsi parce que la situation idéale serait celle où la lettre la plus fréquente de `M0`, soit `t`, proviendrait de la lettre la plus fréquente de l'alphabet, soit `e`[30], et de même pour les lettres suivantes, autrement dit, idéalement, on mettrait en correspondance les chaînes `Aideal` et `Cideal`[31]. *Nous aurions ainsi un algorithme qui décode automatiquement les messages codés par substitution de lettres.* Évidemment, ça ne marche pas[32], ce que nous allons constater sur notre exemple :

Listing 15 – Suite de `Commandes.sce`

```
Aideal='esantirulodcpmqvgfbhxyjzkw';
Cideal=L1+'cnq';//'cnq' trouve a la main. Concatenation de chaines.
exec('Chemin de la fonction CoDec.sci', -1)
Mideal=CoDec(M0, Cideal, Aideal);
afficher('Mideal=');
afficher(Mideal);
```

On obtient `Mideal` qui est incompréhensible et ne peut être le message original :

Listing 16 – Cas idéal et désespéré à la fois

```
Mideal=
 ioesnoeporsxerteerulpeetaenueuenetrplrurtemissiltspaniaoedesnrtnlrdbeanl
 rnprisgrioesndeitnruetliuecexrclenxlrecrsahlpbltecesyadronescltniotecequ
 ainpasaqliufeslitcrmlitspasarnatngresamainuisecesihoatvresrtanlrnplrudlm
 mrtigreuaqedsesdlooevresurssesIrameuidaitsioaruacegrlislddrpeuprisgreslt
 pulvuammecenuaqaiodlmpnafioisecetlmfuersesehpeuietdesaitsioexerteasnulta
 rneseuadlttedneartenafoenneplrudlooedneucesclteesueoaniqesaoapbjsilolvi
 ebrmaitedesenrcesseuqiultntlnammetnaceyitiuceyadltnuespuediseenaoltvneu
 meoesfeslitsaoimetnaiuescesasnultarnesoluscesqlosceoltvrecrueeneodeorigr
 iplruuaxlitcuemausrtearnuemissiltdltsisneuaanesneuoanbeluieceoavuaqinani
 ltseoltoagreooeoavuaqinearteityoretdesruoedlroemetncrnempsplrumesrueuoe
 nempsaqedrteehadninrceenrtesnafioineitevaoeertebluolveanlmigretlrqeooese
 uaitsnaooeeaoehneuieruceoasnaniltlufinaoedenneehpeuietdepeumennuacesjtd
 bultiseuaqedrteporsvuatcepuedisiltoessjsnemesceplsinilttemetnpausaneooin
 e
```

Il faut donc se résoudre à procéder plus systématiquement.

• On a déjà dit que l'on peut raisonnablement supposer que `e -> t`.

• De même, on peut considérer que `s -> l` est à peu près sûr.

• Ensuite, nous ferons le pari très risqué que `a -> a`. C'est risqué parce qu'en fait les fréquences de la troisième à la septième lettres[33] dans `M0` sont assez voisines. On parie que `a -> a`, mais ça pourrait être `a -> m` ou `a -> f` ou `a -> o` ou `a -> w` ou pis. Le risque de perte de temps est donc grand.

• Pour améliorer la lisibilité des résultats, on écrira en majuscules les lettres considérées comme déchiffrées.

• On fait donc `e -> T`, `s -> L`, `a -> A`, ce que traduit la clef `C1` ci-dessous, fabriquée à la main à partir de `Alphabet`.

29. On aurait pu écrire un script *ad hoc* pour calculer `Cideal` automatiquement.
30. ce que nous notons ci-dessous `e -> t`
31. On rappelle que `Aideal` est l'alphabet rangé dans l'ordre des fréquences décroissantes.
32. ça se saurait !
33. rangées suivant les fréquences décroissantes

• Le message obtenu [34] est noté `M1`.

Listing 17 – Suite de `Commandes.sce`

```
C1='AbcdefghijkSmnopqrsEuvwxyz';
M1=CoDec(M0,Alphabet,C1);
afficher('M1=');
afficher(M1);
```

Listing 18 – M1

```
M1=
 osESmsEhswSpEwfEEwkghEEfAEmkEkEmEfwhgwkwfEdoSSogfShAmoAsEeESmwfmgweiEAmg
 wmhwoSjwosESmeEofmwkEfgokErEpwrgEmpgwErwSAvghigfErESyAewsmESrgfmosfErExk
 AomhASAxgokzESgofrwdgofShASAwmAfmjwESAdAomkoSErESovsAfuwESwfAmgwmhgwkegd
 dwfojwEkAxEeSESegssEuwESkwSSESgwAdEkoeAofSosAwkArEjwgoSgeewhEkhwoSjwESgf
 hkgukAddErEmkAxAosegdhmAzosoSErEfgdzkEwSESEvhEkoEfeESAofSosEpEwfEASmkgfA
 wmESEkAegffEemEAwfEmAzsEmmEhgwkegssEemEkrESrgffEESkEsAmoxESAsAhibSogsguo
 EiwdAofEeESEmwrESSEkxokgfmfgmAddEfmArEyofokrEyAegfmkEShkEeoSEEmAsgfumEkd
 EsESzESgofSAsodEfmAokESrESASmkgfAwmESsgkSrESxgsSrEsgfuwErwkEEmEseEswojwo
 hgwkkApgofrkEdAkSwfEAwmkEdoSSogfegfSoSmEkAAmESmEksAmiEgkoErEsAukAxomAmog
 fSEsgfsAjwEssEsAukAxomEAwfEofyswEfeESwksEegwsEdEfmrwmEdhShgwkdESwkEksEmE
 dhSAxEewfEEvAemomwrEEmwfESmAzosomEofEuAsEEwfEigksguEAmgdojwEfgwxEssESEkA
 ofSmAssEEAsEvmEkoEwkrEsASmAmogfgkzomAsEeEmmEEvhEkoEfeEhEkdEmmkArESbfeik
 gfoSEkAxEewfEhswSukAfrEhkEeoSogfsESSbSmEdESrEhgSomogffEdEfmhAkSAmEssomE
```

On n'y voit rien de rédhibitoire. Essayons maintenant de débusquer la lettre qui donnera `N`, qui est la quatrième lettre la plus fréquente de l'alphabet. En se guidant avec `Cideal`, on commence par tester `m -> N`, puis si nécessaire, on essaiera `f -> N`, `o -> N`, `w -> N`, *etc.*

Listing 19 – Suite de `Commandes.sce`

```
//Essai de a->A et de m->N.
C2='AbcdefghijkSNnopqrsEuvwxyz';
M2=CoDec(M0,Alphabet,C2);
afficher('M2=');
afficher(M2);
```

Listing 20 – Cul de sac

```
 M2=
 osESNsEhswSpEwfEEwkghEEfAENkEkENEfwhgwkwfEdoSSogfShANoAsEeESNwfNgweiEANg
 wNhwoSjwosESNeEofNwkEfgokErEpwrgENpgwErwSAvghigfErESyAewsNESrgfNosfErExk
 AoNhASAxgokzESgofrwdgofShASAwNAfNjwESAdAoNkoSErESovsAfuwESwfANgwNhgwkegd
 dwfojwEkAxEeSESegssEuwESkwSSESgwAdEkoeAofSosAwkArEjwgoSgeewhEkhwoSjwESgf
 hkgukAddErENkAxAosegdhNAzosoSErEfgdzkEwSESEvhEkoEfeESAofSosEpEwfEASNkgfA
 wNESEkAegffEeNEAwfENAzsENNEhgwkegssEeNEkrESrgffEESkEsANoxESAsAhibSogsguo
 EiwdAofEeESENwrESSEkxokgfNfgNAddEfNArEyofokrEyAegfNkEShkEeoSEENAsgfuNEkd
 EsESzESgofSAsodEfNAokESrESASNkgfAwNESsgkSrESxgsSrEsgfuwErwkEENEseEswojwo
 hgwkkApgofrkEdAkSwfEAwNkEdoSSogfegfSoSNEkAANESNEksANiEgkoErEsAukAxoNANog
 fSEsgfsAjwEssEsAukAxoNEAwfEofyswEfeESwksEegwsEdEfNrwNEdhShgwkdESwkEksENE
 dhSAxEewfEEvAeNoNwrEENwfESNAzosoNEofEuAsEEwfEigksguEANgdojwEfgwxEssESEkA
```

34. sur la voie du décodage ?

```
ofSNAssEEAsEvNEkoEwkrEsASNANogfgkzoNAsEeENNEEvhEkoEfeEhEkdENNkArESbfeikg
foSEkAxEewfEhswSukAfrEhkEeoSogfsESSbSNEdESrEhgSoNogffEdEfNhAkSANEssoNE
```

Nous ne voyons aucun groupement de lettres majuscules qui fait penser à un mot, mais plutôt des groupements suspects. Nous rejetterons donc la tentative `m -> N`. Testons à sa place `f -> N`, ce qui va conduire à `C3` et `M3`.

Listing 21 – Suite de `Commandes.sce`

```
//Essai de a->A et de f->N.
C3='AbcdeNghijkSmnopqrsEuvwxyz';
M3=CoDec(M0,Alphabet,C3);
afficher('M3=');
afficher(M3);
```

ce qui donne

Listing 22 – M3

```
M3=
osESmsEhswSpEwNEEwkghEENAEmkEkEmENwhgwkwNEdoSSogNShAmoAsEeESmwNmgweiEAmg
wmhwoSjwosESmeEoNmwkENgokErEpwrgEmpgwErwSAvghigNErESyAewsmESrgNmosNErExkA
omhASAxgokzESgoNrwdgoNShASAwmANmjwESAdAomkoSErESovsANuwESwNAmgwmhgwkegdd
wNojwEkAxEeSESegssEuwESkwSSESgwAdEkoeAoNSosAwkArEjwgoSgeewhEkhwoSjwESgNhk
gukAddErEmkAxAosegdhmAzosoSErENgdzkEwSESEvhEkoENeESAoNSosEpEwNEASmkgNAwm
ESEkAegNNEemEAwNEmAzsEmmEhgwkegssEemEkrESrgNNEESkEsAmoxESAsAhibSogsguoEiw
dAoNEeESEmwrESSEkxokgNmNgmAddENmArEyoNokrEyAegNmkEShkEeoSEEmAsgNumEkdEsE
SzESgoNSAsodENmAokESrESASmkgNAwmESsgkSrESxgsSrEsgNuwErwkEEmEseEswojwohgwk
kApgoNrkEdAkSwNEAwmkEdoSSogNegNSoSmEkAAmESmEksAmiEgkoErEsAukAxomAmogNSEsg
NsAjwEssEsAukAxomEAwNEoNyswENeESwksEegwsEdENmrwmEdhShgwkdESwkEksEmEdhSAxE
ewNEEvAemomwrEEmwNESmAzosomEoNEuAsEEwNEigksguEAmgdojwENgwxEssESEkAoNSmAss
EEAsEvmEkoEwkrEsASmAmogNgkzomAsEeEmmEEvhEkoENeEhEkdEmmkArESbNeikgNoSEkAxE
ewNEhswSukANrEhkEeoSogNsESSbSmEdESrEhgSomogNNEdENmhAkSAmEssomE
```

Dans `M3`, la séquence `ESrgNNEES` invite à tenter `r -> D` et `g -> O`. On introduit ainsi `C4` et `M4` :

Listing 23 – Suite de `Commandes.sce`

```
C4='AbcdeNOhijkSmnopqDsEuvwxyz';
M4=CoDec(M0,Alphabet,C4);
afficher('M4=');
afficher(M4);
```

ce qui donne :

Listing 24 – M4

```
M4=
osESmsEhswSpEwNEEwkOhEENAEmkEkEmENwhOwkwNEdoSSoONShAmoAsEeESmwNmOweiEAmOw
mhwoSjwosESmeEoNmwkENOokEDEpwDOEmpOwEDwSAvOhiONEDESyAewsmESDONmosNEDExkAo
mhASAxOokzESOoNDwdOoNShASAwmANmjwESAdAomkoSEDESovsANuwESwNAmOwmhOwkeOddwN
ojwEkAxEeSESeOssEuwESkwSSESOwAdEkoeAoNSosAwkADEjwOoSOeewhEkhwoSjwESONhkOu
kAddEDEmkAxAoseOdhmAzosoSEDENOdzkEwSESEvhEkoENeESAoNSosEpEwNEASmkONAwmESE
kAeONNEemEAwNEmAzsEmmEhOwkeOssEemEkDESDONNEESkEsAmoxESAsAhibSoOsOuoEiwdAo
NEeESEmwDESSEkxokONmNOmAddENmADEyoNokDEyAeONmkEShkEeoSEEmAsONumEkdEsESzES
OoNSAsodENmAokESDESASmkONAwmESsOkSDESxOsSDEsONuwEDwkEEmEseEswojwohOwkkApO
```

```
oNDkEdAkSwNEAwmkEdoSSoONeONSoSmEkAAmESmEksAmiEOkoEDEsAukAxomAmoONSEsONsAj
wEssEsAukAxomEAwNEoNyswENeESwksEeOwsEdENmDwmEdhShOwkdESwkEksEmEdhSAxEewNE
EvAemomwDEEmwNESmAzosomEoNEuAsEEwNEiOksOuEAmOdojwENOwxEssESEkAoNSmAssEEAs
EvmEkoEwkDEsASmAmoONOkzomAsEeEmmEEvhEkoENeEhEkdEmmkADESbNeikONoSEkAxEewNE
hswSukANDEhkEeoSoONsESSbSmEdESDEhOSomoONNEdENmhAkSAmEssomE
```

Si nous sommes sur la bonne voie, nous avons déjà décodé 6 lettres, à savoir `E,S,A,N,O,D`. Recherchons d'où proviennent les lettres `t` et `i` qui sont les lettres les plus fréquentes de l'alphabet dans les lettres encore manquantes. Essayons `m -> T` (en tenant compte de `Cideal`)

Listing 25 – Suite de `Commandes.sce`

```
C5='AbcdeNOhijkSTnopqDsEuvwxyz';
M5=CoDec(M0, Alphabet ,C5);
afficher('M5=');
afficher(M5);
```

ce qui donne `M5` :

Listing 26 – M5

```
 M5=
osESTsEhswSpEwNEEwkOhEENAETkEkETENwhOwkwNEdoSSoONShATOAsEeESTwNTOweiEATOw
ThwoSjwosESTeEoNTwkENOokEDEpwDOETpOwEDwSAvOhiONEDESyAewsTESDONTosNEDExkAo
ThASAxOokzESOoNDwdOoNShASAwTANTjwESAdAoTkoSEDESovsANuwESwNATOwThOwkeOddwN
ojwEkAxEeSESeOssEuwESkwSSESOwAdEkoeAoNSosAwkADEjwOoSOeewhEkhwoSjwESONhkOu
kAddEDETkAxAoseOdhTAzosoSEDENOdzkEwSESEvhEkoENeESAoNSosEpEwNEASTkONAwTESE
kAeONNEeTEAwNETAzsETTEhOwkeOssEeTEkDESDONNEESkEsATOxESAsAhibSoOsOuoEiwdAo
NEeESETwDESSEkxokONTNOTAddENTADEyoNokDEyAeONTkEShkEeoSEETAsONuTEkdEsESzES
OoNSAsodENTAokESDESASTkONAwTESsOkSDESxOsSDEsONuwEDwkEETEseEswojwohOwkkApO
oNDkEdAkSwNEAwTkEdoSSoONeONSoSTEkAATESTEksATiEOkoEDEsAukAxoTATOONSEsONsAj
wEssEsAukAxoTEAwNEoNyswENeESwksEeOwsEdENTDwTEdhShOwkdESwkEksETEdhSAxEewNE
EvAeToTwDEETwNESTAzosoTEoNEuAsEEwNEiOksOuEATOdojwENOwxEssESEkAoNSTAssEEAs
EvTEkoEwkDEsASTATOONOkzoTAsEeETTEEvhEkoENeEhEkdETTkADESbNeikONoSEkAxEewNE
hswSukANDEhkEeoSoONsESSbSTEdESDEhOSoToONNEdENThAkSATEssoTE
```

Interrompons la recherche du `i` car la séquence `ONTNOTAddENTADE` invite à faire `d -> M`.

Listing 27 – Suite de `Commandes.sce`

```
C6='AbcMeNOhijkSTnopqDsEuvwxyz';
M6=CoDec(M0, Alphabet ,C6);
afficher('M6=');
afficher(M6);
```

Listing 28 – M6

```
 M6=
osESTsEhswSpEwNEEwkOhEENAETkEkETENwhOwkwNEMoSSoONShATOAsEeESTwNTOweiEATOw
ThwoSjwosESTeEoNTwkENOokEDEpwDOETpOwEDwSAvOhiONEDESyAewsTESDONTosNEDExkAo
ThASAxOokzESOoNDwMOoNShASAwTANTjwESAMAoTkoSEDESovsANuwESwNATOwThOwkeOMMw
NojwEkAxEeSESeOssEuwESkwSSESOwAMEkoeAoNSosAwkADEjwOoSOeewhEkhwoSjwESONhkO
ukAMMEDETkAxAoseOMhTAzosoSEDENOMzkEwSESEvhEkoENeESAoNSosEpEwNEASTkONAwTES
EkAeONNEeTEAwNETAzsETTEhOwkeOssEeTEkDESDONNEESkEsATOxESAsAhibSoOsOuoEiwMA
oNEeESETwDESSEkxokONTNOTAMMENTADEyoNokDEyAeONTkEShkEeoSEETAsONuTEkMEsESzE
SOoNSAsoMENTAokESDESASTkONAwTESsOkSDESxOsSDEsONuwEDwkEETEseEswojwohOwkkAp
```

```
OoNDkEMAkSwNEAwTkEMoSSoONeONSoSTEkAATESTEksATiEOkoEDEsAukAxoTAToONSEsONs
AjwEssEsAukAxoTEAwNEoNyswENeESwksEeOwsEMENTDwTEMhShOwkMESwkEksETEMhSAxEew
NEEvAeToTwDEETwNESTAzosoTEoNEuAsEEwNEiOksOuEATOMojwENOwxEssESEkAoNSTAssEE
AsEvTEkoEwkDEsASTAToONOkzoTAsEeETTEEvhEkoENeEhEkMETTkADESbNeikONoSEkAxEew
NEhswSukANDEhkEeoSoONsESSbSTEMESDEhOSoToONNEMENThAkSATEssoTE
```

Nous avons maintenant décodé 8 lettres, c'est à peu près sûr. Les séquences `SbSTEMES` et `SATEssoTE` invitent a essayer `b -> Y`, `s -> L` et `o -> i`.

Listing 29 – Suite de `Commandes.sce`

```
C7='AYcMeNOhijkSTnIpqDLEuvwxyz';
M7=CoDec(M0, Alphabet, C7);
afficher('M7=');
afficher(M7);
```

On obtient ainsi :

Listing 30 – M7

```
M7=
ILESTLEhLwSpEwNEEwkOhEENAETkEkETENwhOwkwNEMISSIONShATIALEeESTwNTOweiEATOw
ThwISjwILESTeEINTwkENOIkEDEpwDOETpOwEDwSAvOhiONEDESyAewLTESDONTILNEDExkAI
ThASAxOIkzESOINDwMOINShASAwTANTjwESAMAITkISEDESIvLANuwESwNATOwThOwkeOMMw
NIjwEkAxEeSESeOLLEuwESkwSSESOwAMEkIeAINSILAwkADEjwOISOeewhEkhwISjwESONhkO
ukAMMEDETkAxAILeOMhTAzILISEDENOMzkEwSESEvhEkIENeESAINSILEpEwNEASTkONAwTES
EkAeONNEeTEAwNETAzLETTEhOwkeOLLEeTEkDESDONNEESkELATIxESALAhiYSIOLOuIEiwMA
INEeESETwDESSEkxIkONTNOTAMMENTADEyINIkDEyAeONTkEShkEeISEETALONuTEkMELESzE
SOINSALIMENTAIkESDESASTkONAwTESLOkSDESxOLSDELONuwEDwkEETELeELwIjwIhOwkkAp
OINDkEMAkSwNEAwTkEMISSIONeONSISTEkAATESTEkLATiEOkIEDELAukAxITATIONSELONLA
jwELLELAukAxITEAwNEINyLwENeESwkLEeOwLEMENTDwTEMhShOwkMESwkEkLETEMhSAxEewN
EEvAeTITwDEETwNESTAzILITEINEuALEEwNEiOkLOuEATOMIjwENOwxELLESEkAINSTALLEEA
LEvTEkIEwkDELASTATIONOkzITALEeETTEEvhEkIENeEhEkMETTkADESYNeikONISEkAxEewN
EhLwSukANDEhkEeISIONLESSYSTEMESDEhOSITIONNEMENThAkSATELLITE
```

La séquence `SYSTEMESDEhOSITIONNEMENThAkSATELLITE` invite à essayer `h -> P`, `k -> R` ; la séquence `LEhLwS` invite ensuite à tester `w->U`.

Listing 31 – Suite de `Commandes.sce`

```
C8='AYcMeNOPijRSTnIpqDLEuvUxyz';
M8=CoDec(M0, Alphabet, C8);
afficher('M8=');
afficher(M8);
```

Listing 32 – M8

```
M8=
ILESTLEPLUSpEUNEEUROPEENAETRERETENUPOURUNEMISSIONSPATIALEeESTUNTOUeiEATOU
TPUISjUILESTeEINTURENOIREDEpUDOETpOUEDUSAvOPiONEDESyAeULTESDONTILNEDExRAI
TPASAxOIRzESOINDUMOINSPASAUTANTjUESAMAITRISEDESIvLANuUESUNATOUTPOUReOMMU
NIjUERAxEeSESeOLLEuUESRUSSESOUAMERIeAINSILAURADEjUOISOeeUPERPUISjUESONPRO
uRAMMEDETRAxAILeOMPTAzILISEDENOMzREUSESEvPERIENeESAINSILEpEUNEASTRONAUTES
ERAeONNEeTEAUNETAzLETTEPOUReOLLEeTERDESDONNEESRELATIxESALAPiYSIOLOuIEiUMA
INEeESETUDESSERxIRONTNOTAMMENTADEyINIRDEyAeONTRESPREeISEETALONuTERMELESzE
```

```
SOINSALIMENTAIRESDESASTRONAUTESLORSDESxOLSDELONuUEDUREETELeELUIjUIPOURRAp
OINDREMARSUNEAUTREMISSIONeONSISTERAATESTERLATiEORIEDELAuRAxlTATIONSELONLA
jUELLELAuRAxlTEAUNEINyLUENeESURLEeOULEMENTDUTEMPSPOURMESURERLETEMPSAxEeUN
EEvAeTITUDEETUNESTAzILITEINEuALEEUNEiORLOuEATOMIjUENOUxELLESERAINSTALLEEA
LEvTERIEURDELASTATIONORzITALEeETTEEvPERIENeEPERMETTRADESYNeiRONISERAxEeUN
EPLUSuRANDEPREeISIONLESSYSTEMESDEPOSITIONNEMENTPARSATELLITE
```

C'est facile maintenant. `PLUSpEUNE` $\Longrightarrow$ (`p -> j`), `DENOMzREUSESEvPERIENeES` $\Longrightarrow$ (`z -> B`), (`v -> X`) et (`e -> C`), `ETUDESSERxIRONT` $\Longrightarrow$ (`x -> v`), `ADEyINIRDE` $\Longrightarrow$ (`y->F`).

Listing 33 – Suite de `Commandes.sce`

```
C9='AYcMCNOPijRSTnIJqDLEuXUVFB';
M9=CoDec(M0, Alphabet, C9);
afficher('M9=');
afficher(M9);
```

Listing 34 – M9

```
M9=
ILESTLEPLUSJEUNEEUROPEENAETRERETENUPOURUNEMISSIONSPATIALECESTUNTOUCiEATOU
TPUISjUILESTCEINTURENOIREDEJUDOETJOUEDUSAXOPiONEDESFACULTESDONTILNEDEVRAI
TPASAVOIRBESOINDUMOINSPASAUTANTjUESAMAITRISEDESIXLANuUESUNATOUTPOURCOMMUN
IjUERAVECSESCOLLEuUESRUSSESOUAMERICAINSILAURADEjUOISOCCUPERPUISjUESONPROu
RAMMEDETRAVAILCOMPTABILISEDENOMBREUSESEXPERIENCESAINSILEJEUNEASTRONAUTESE
RACONNECTEAUNETABLETTEPOURCOLLECTERDESDONNEESRELATIVESALAPiYSIOLOuIEiUMAI
NECESETUDESSERVIRONTNOTAMMENTADEFINIRDEFACONTRESPRECISEETALONuTERMELESBES
OINSALIMENTAIRESDESASTRONAUTESLORSDESVOLSDELONuUEDUREETELCELUIjUIPOURRAJO
INDREMARSUNEAUTREMISSIONCONSISTERAATESTERLATiEORIEDELAuRAVITATIONSELONLAj
UELLELAuRAVITEAUNEINFLUENCESURLECOULEMENTDUTEMPSPOURMESURERLETEMPSAVECUNE
EXACTITUDEETUNESTABILITEINEuALEEUNEiORLOuEATOMIjUENOUVELLESERAINSTALLEEAL
EXTERIEURDELASTATIONORBITALECETTEEXPERIENCEPERMETTRADESYNCiRONISERAVECUNE
PLUSuRANDEPRECISIONLESSYSTEMESDEPOSITIONNEMENTPARSATELLITE
```

Enfin `TOUCiEATOUT` $\Longrightarrow$ (`i -> h`), `PUISjUILEST` $\Longrightarrow$ (`j -> Q`), `LANuUES` $\Longrightarrow$ (`u -> G`).

Listing 35 – Suite de `Commandes.sce`

```
C10='AYcMCNOPHQRSTnIJqDLEGXUVFB';
M10=CoDec(M0, Alphabet, C10);
afficher('M10=');
afficher(M10);
```

ce qui termine le décodage du message `M0`[35] :

Listing 36 – Message décodé

```
 M10=
ILESTLEPLUSJEUNEEUROPEENAETRERETENUPOURUNEMISSIONSPATIALECESTUNTOUCHEATOU
TPUISQUILESTCEINTURENOIREDEJUDOETJOUEDUSAXOPHONEDESFACULTESDONTILNEDEVRAI
TPASAVOIRBESOINDUMOINSPASAUTANTQUESAMAITRISEDESIXLANGUESUNATOUTPOURCOMMUN
IQUERAVECSESCOLLEGUESRUSSESOUAMERICAINSILAURADEQUOISOCCUPERPUISQUESONPROG
RAMMEDETRAVAILCOMPTABILISEDENOMBREUSESEXPERIENCESAINSILEJEUNEASTRONAUTESE
RACONNECTEAUNETABLETTEPOURCOLLECTERDESDONNEESRELATIVESALAPHYSIOLOGIEHUMAI
```

35. Extrait un peu modifié de [8]

NECESETUDESSERVIRONTNOTAMMENTADEFINIRDEFACONTRESPRECISEETALONGTERMELESBE
SOINSALIMENTAIRESDESASTRONAUTESLORSDESVOLSDELONGUEDUREETELCELUIQUIPOURRAJ
OINDREMARSUNEAUTREMISSIONCONSISTERAATESTERLATHEORIEDELAGRAVITATIONSELONLA
QUELLELAGRAVITEAUNEINFLUENCESURLECOULEMENTDUTEMPSPOURMESURERLETEMPSAVECUN
EEXACTITUDEETUNESTABILITEINEGALEEUNEHORLOGEATOMIQUENOUVELLESERAINSTALLEEA
LEXTERIEURDELASTATIONORBITALECETTEEXPERIENCEPERMETTRADESYNCHRONISERAVECUN
EPLUSGRANDEPRECISIONLESSYSTEMESDEPOSITIONNEMENTPARSATELLITE

6.2 Commentaires

• On trouve d'autres exemples sur la toile. Ainsi, sur le site de *cryptage.org*, il y a un bel exemple entièrement traité et convaincant, voir [11].

• On trouve sur la toile des textes à décoder [36], voir par exemple [4].

• La méthode des fréquences a d'autres ressources comme l'utilisation des fréquences de bigrammes [37].

• Pour échapper à la menace de décodage sauvage par la méthode des fréquences, on peut la rendre inopérante en faisant intervenir dans le chiffrement plusieurs codages par substitution (cf. le chiffrement de Hill, [1] ou la machine Enigma, voir [7]).

7 Conclusion

7.1 Idées d'activités sur le thème du déchiffrement de messages codés

• On peut évidemment proposer des activités d'algorithmique où le déchiffrement est un simple prétexte [38]. Par exemple, une substitution de lettres ayant été définie par une clef quelconque donnée, demander de programmer une fonction *scilab* jouant le rôle de la fonction `CoDec` ci-dessus, l'utiliser pour coder un texte donné, puis proposer le décodage d'un autre texte donné codé avec la même substitution de lettres [39].

• Si l'on veut vraiment mettre en œuvre la méthode des fréquences, on devra se limiter au lycée au cas qui marche, c'est à dire au déchiffrement d'un texte codé par décalage de pas inconnu, cf. 5.2.2.

7.2 Critique l'utilisation de la méthode des fréquences au lycée

La principale critique que l'on peut faire à la méthode des fréquences est qu'elle ne propose aucun modèle, c'est à dire aucun cadre mathématique dans lequel on pourrait justifier les calculs faits. Pire, quand on affirme que la méthode marche d'autant mieux que le message à décoder est plus long, on sent la présence importune de la loi des grands nombres [40]. Si l'on pouvait prétendre que l'on écrit les messages codés en tirant leurs lettres successives au hasard dans l'alphabet suivant les fréquences communément admises, les tirages étant indépendants les uns des autres, cela définirait un modèle et on pourrait invoquer la fameuse loi. Mais, c'est bien évidemment absurde. Malgré tout, le risque de confusion est grand.

La méthode des fréquences ne se justifie pas mathématiquement. Ça marche [41] quand on a du temps, de l'intuition et si possible de la chance.

36. qui ont été codés par substitution de lettres
37. Bigramme signifie couple de lettres. On peut s'intéresser à des bigrammes particuliers : bigrammes de voyelles, de consonnes, consonnes redoublées, *etc*, voir [3].
38. Par exemple, dans [1], il y avait une belle activité, difficile, sur le chiffrement de Hill, mais c'était surtout un problème de calcul matriciel et d'arithmétique.
39. ce qui nécessitera de comprendre que la fonction *scilab* écrite précédemment fonctionne aussi pour déchiffrer
40. qui n'est jamais invoquée explicitement
41. ce qui est agaçant !

Références

[1] Chiffrement de Hill *Ressources pour la classe terminale et technologique, Mathématiques, Série S, Enseignement de spécialité*, BO spécial n° 8 du 13 octobre 2011, I, paragraphe LIV, *nrf*, Pléiade, Gallimard, 1968.

[2] Chiffrement par substitution.
http://iml.univ-mrs.fr/%7Erodier/Cours/Substitution.tpts.pdf

[3] Fréquences des lettres, bigrammes, trigrammes, quadrigrammes en français, Site de Bibm@th.net.
http://www.bibmath.net/crypto/index.php?action=affiche&quoi=chasseur/frequences_francais

[4] Fréquences des lettres en français, Site de Apprendre-en-ligne.net.
http://www.apprendre-en-ligne.net/crypto/stat/francais.html

[5] Fréquences des lettres et polygrammes dans la langue française, Site de Bibm@th.net.
http://bibmath.net/forums/viewtopic.php?id=3767

[6] Historiens romains. Historiens de la République *II César La guerre civile* I, paragraphe LIV, *nrf*, Pléiade, Gallimard, 1968.

[7] Lapôtre Pierre & Raymond Moché, La machine Enigma, activité
http//www.gradus-ad-mathematicam.fr/TS_Algorithmique4.htm

[8] Monier Pierre *Thomas Pesquet, le numéro 10 de l'équipe spatiale française*, article publié le 21 novembre 2015 sur le site de *L'usine nouvelle*, section Spatial.

[9] Site de scilab.
http://www.scilab.org/fr/

[10] Site de cryptage.org
http://www.cryptage.org

[11] Tableau des fréquences pour la langue française, Site de cryptage.org.
http://www.cryptage.org/analyse-frequentielle.html

[12] Wikipedia. *American Standard for Information Interchange*
https://fr.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange

[13] Wikipedia. *Chiffrement par décalage*
https://fr.wikipedia.org/wiki/Chiffrement_par_decalage

[14] Wikipedia. *Chiffrement par substitution*
https://fr.wikipedia.org/wiki/Chiffrement_par_substitution