

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use Math::Trig ;
use strict;
```

```
#!/(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort { $a->$b } keys %coor ) {
```

Лекция 3. Введение в квантовую химию

Курс: Молекулярное моделирование в применении к биомолекулам

Головин А.В. ¹

¹МГУ им М.В. Ломоносова, Факультет Биоинженерии и Биоинформатики

Москва, 2013

```
my %qwa=find_quart( $coor{"O"} ); my $qnum=keys %qwa;

if ($qnum >0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^\.*/;/;
$filename=~ s/\.pdb//;
#$filename=$chnum.".".$qnum.".".$filename.".dat";
$filename="$dir".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";

foreach my $m (sort { $a->$b } keys %coor){
my %qartets= %qwa ; #find quart( $coor{$m} );
my %q= find

# foreach my $q { keys %qartets } { print join " ",@{ $qartets{$q} },"\n";

foreach my $q { keys %qartets } {

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

foreach my $res (@{ $qartets{$q} }){

# print "$q $coor{$m}{ $res } {"R"}->x,\n";
$nx=$nx+ $coor{$m}{ $res }{"N9"}->x;
$ny=$ny+ $coor{$m}{ $res }{"N9"}->y;
$nz=$nz+ $coor{$m}{ $res }{"N9"}->z;

$ox=$ox+ $coor{$m}{ $res }{"O6"}->x;
$oy=$oy+ $coor{$m}{ $res }{"O6"}->y;
$oz=$oz+ $coor{$m}{ $res }{"O6"}->z;
$r=$res;
}
}
```

Содержание

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );

#(my %$coor,my $schnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $schnum;
foreach my $r ( sort keys %{$coor{"0"}}){ my $ggg=substr($r,0,1); if ( $ggg ne $sch){ $schnum++; $sch=$ggg } ;
```

```
my %$qwa=find_quart( %$coor{"0"} ); my $sqnum=keys %$qwa;
```

```
if ($sqnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/\./\//;
$filename=~ s/\./\//;
#$filename=$schnum."_"$sqnum."_"$filename.".dat";
$filename="$dir/".$filename.".dat";
print OUT "#INFO chain $schnum qnum $sqnum\n";
```

```
foreach my $m (sort { $a<=>$b } keys %$coor){
my %$qartets= %$qwa; #find_quart( %$coor{$m} );
my %$q= find_q( %$coor{$m} );
```

```
# foreach my $q ( keys %$qartets){ print join " ", @{$qartets{$q}}, "\n";
```

```
foreach my $q ( keys %$qartets){
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}}){
```

```
# print "$q $coor{$m} {$res} {"$r"}->x, "\n";
$nx=$nx+ $coor{$m} {$res} {"$r"}->x;
$ny=$ny+ $coor{$m} {$res} {"$r"}->y;
$nz=$nz+ $coor{$m} {$res} {"$r"}->z;
```

```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

Активные молекулы

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Квантовая химия

```
my ($coor,$schnum)=read_pdb($ARGV[0]);
my $coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch,$schnum;
```

Квантовая химия это применение квантовой механики для исследования химических проблем.

- Экспериментальная квантовая химия связана со спектроскопией (IR,NMR)
- Теоретическая квантовая химия это исследование электронной структуры системы вычислительными методами. Т.е. решение уравнения Шредингера для молекулярной системы. Использование термина *Abinitio* не предполагает использование экспериментальных данных, только основные физические константы.

```
foreach my $q ( keys %qartets ){
    my $qartets=$qartets{$q};
    my $qartets=$qartets{$q};
    # foreach my $q ( keys %qartets ){
    foreach my $q ( keys %qartets ){
        my $qartets=$qartets{$q};
        my $qartets=$qartets{$q};
        #
        print "$q $coor{$$m} {$sres} {"$N"}->x,"$n";
        $nx=$nx+ $coor{$$m} {$sres} {"$N"}->x;
        $ny=$ny+ $coor{$$m} {$sres} {"$N"}->y;
        $nz=$nz+ $coor{$$m} {$sres} {"$N"}->z;

        $ox=$ox+ $coor{$$m} {$sres} {"$O6"}->x;
        $oy=$oy+ $coor{$$m} {$sres} {"$O6"}->y;
        $oz=$oz+ $coor{$$m} {$sres} {"$O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Энергия и вещество

```
#!/(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"O"}}( my $sggg=substr($r,0,1); if ( $sggg ne $ch){ $chnum++; $ch=$sggg } );
```

Размер частицы Свойство частицы Свойство волны

```
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $file="";
$file=$dir."/";
$file=$dir."/";
$file=$dir."/";
$file=$dir."/";
$file=$dir."/";
print "chain $chnum qnum $qnum\n";
open(FILE,">$file");
print OUT "#IN-O chain $chnum qnum $qnum\n";
```

Крупный (ядро)

В основном

Не наблюдается

Средний (электрон)

Частично

Частично

```
foreach my $m (sort {$a<=>$b} keys %coor){
my %qartets = %qartets + $coor{$m};
my %qartets = %qartets + $coor{$m};
```

Малый (фонон)

Не много

В основном

```
# foreach my $q ( keys %qartets){ print join " ",@{$qartets{$q}}, "\n";
```

```
foreach my $q ( keys %qartets){
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}}){
```

```
# print "$q $coor{$m} {$res} {"N"}->x,\n";
```

```
$nx=$nx+ $coor{$m} {$res} {"N9"}->x;
```

```
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
```

```
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
```

```
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
```

```
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Волновая функция

```
my ($my %coor,$my $chain) = read_pdb($ARGV[0]);
my $chain = read_pdb($ARGV[0]);
```

Wikipedia* :

Волновая функция — комплекснозначная функция, используемая для описания чистого квантового состояния системы. Обычно функция имеет комплексные значения, а для одной частицы это функция пространства и времени. Изменение волновой функции сравнимо с поведением волны.

Физический смысл волновой функции заключается в том, что согласно копенгагенской интерпретации квантовой механики плотность вероятности нахождения частицы в данной точке пространства в данный момент времени считается равной квадрату абсолютного значения волновой функции этого состояния в координатном представлении.

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Волновая функция:

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $ch ){ $chnum++; $ch=$ggg } ;

```

```
my %qwa=find_quart( %coor{"0"} ); my $qnum=keys %qwa;
```

```

if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/~/~/";
$filename="-- s/\.pdb//";
#$filename=$chnum.".".$qnum.".".$filename.".dat";
$filename="$dir".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";

```

```

foreach my $m ( sort { $a-<=>$b } keys %coor ){
my %qartets=@{ $qwa{ $m } }; #find_quart( %coor{ $m } );
my %q= %qartets{ $m };

```

```
# foreach my $q ( sort { $a-<=>$b } keys %qartets ){ print join " ",@{ $qartets{ $q } },"\n";
```

```
foreach
```

```
my $m
```

```
my $o
```

```
my $
```

```
foreach
```

```
#
```

```
print "$q $coor{ $m } { $res } { "N7" }->x, "\n";
```

```
$nx=$nx+ $coor{ $m } { $res } { "N9" }->x;
```

```
$ny=$ny+ $coor{ $m } { $res } { "N9" }->y;
```

```
$nz=$nz+ $coor{ $m } { $res } { "N9" }->z;
```

```
$ox=$ox+ $coor{ $m } { $res } { "O6" }->x;
```

```
$oy=$oy+ $coor{ $m } { $res } { "O6" }->y;
```

```
$oz=$oz+ $coor{ $m } { $res } { "O6" }->z;
```

Траектории гармонического осциллятора:

- В классической механике A-B.
- В квантовой механике C-H, реальная часть волновой функции шарика показана синим, красным показана комплексная часть.
- C-F это примеры стационарного состояния.

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Уравнение Шредингера

Вариант движения электрона (вектор r) с учётом времени в присутствии внешнего электрического поля V . Это не релятивистский вариант для одной частицы.

```
if ($qnum > 0) {
  #system("mv $ARGV[1]");
  my $fdir=$ARGV[1];
  #files($fdir);
  my $filename="pdbr";
  #filename=$qnum." ".$qnum."7";
  $filename="$fdir"."$filename"."dat";
  print "$filename\n";
  #print "QOUT > $filename";
  foreach my $m (sort { $a<=>$b } keys %coor) {
    my %qartets= %qwa ; #find quart { $coor {$m} };
    my %q= find of $coor{$m} ;
    # foreach my $q ( keys %qartets) { print join " ", @qartets{$q} ; \n
    foreach my $q ( keys %qartets) {
      my $nx; my $ny; my $nz;
      my $ox; my $oy; my $oz;
      my $r;
      foreach my $res ( @ { $qartets{$q} } ) {
        print "$q $coor{$m} {$res} {"
        $nx=$nx+ $coor{$m} {$res} {"N9"};
        $ny=$ny+ $coor{$m} {$res} {"N9"};
        $nz=$nz+ $coor{$m} {$res} {"N9"}->z;
        $ox=$ox+ $coor{$m} {$res} {"O6"}->x;
        $oy=$oy+ $coor{$m} {$res} {"O6"}->y;
        $oz=$oz+ $coor{$m} {$res} {"O6"}->z;
      }
    }
  }
}
```

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = \left[\frac{-\hbar^2}{2m} \nabla^2 + V(\mathbf{r}, t) \right] \Psi(\mathbf{r}, t); \quad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

\hbar - постоянная планка; Ψ - волновая функция.

Если V не изменяется со временем то для частицы с энергией E :

$$\Psi(r, t) = \psi(r)T(t)$$

$$\left[\frac{-\hbar^2}{2m} \nabla^2 + V \right] \Psi(r) = E\Psi(r)$$

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Уравнение Шредингера

```
my ($my $coor, my $schnum)=read_pdb($ARGV[0]);
```

```
my %coor=read_pdb($ARGV[0]);
```

Итак обзовем оператором H (Гамильтониан):

```
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=subst($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg } };
```

```
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;
```

$$H = \frac{-\hbar^2}{2m} \nabla^2 + V$$

```
if ($qnum > 0){
```

```
  #system("mkdir $ARGV[1]");
```

```
  my $filename=$ARGV[0];
```

```
  $filename="-- s/\^.*\$/";
```

```
  $filename="-- s/\^.*\$/";
```

```
  # $filename=$schnum." ".$qnum." ".$filename." .dat";
```

```
  $filename="$dir".$filename.".dat";
```

```
  print OUT "\n";
```

```
  print OUT "chain $schnum qnum $qnum \n";
```

```
  print OUT "chain $schnum qnum $qnum \n";
```

```
  foreach my $m ( sort { $a-<=>$b } keys %coor ){
```

```
    my %qartets = %qwa; #find_quart( $coor{ $m } );
```

```
    my %q = find_q( $coor{ $m } );
```

$$H\Psi = E\Psi$$

```
  #   foreach my $q { keys %qartets } { print join " ", @{$qartets{$q}}, "\n";
```

Для решения этого уравнения надо найти значения E и волновой

функции. Это уравнение относится к типу дифференциальных

уравнений с собственными значениями, где оператор действующий на

функцию возвращает произведение скалярной величины на функцию.

```
  $nx=$nx+ $coor{ $m } { $res } {"N9"}->x;
```

```
  $ny=$ny+ $coor{ $m } { $res } {"N9"}->y;
```

```
  $nz=$nz+ $coor{ $m } { $res } {"N9"}->z;
```

```
  $ox=$ox+ $coor{ $m } { $res } {"O6"}->x;
```

```
  $oy=$oy+ $coor{ $m } { $res } {"O6"}->y;
```

```
  $oz=$oz+ $coor{ $m } { $res } {"O6"}->z;
```



```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Операторы

```
#[my %coor,my $chnum]=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
```

Ожидаемое значение (можно рассматривать как среднее значение) какого либо свойства: энергии, положения, линейного момента, можно определить с помощью оператора.

Пример: гамильтониан это оператор для энергии можно сказать, что зная волновую функцию:

$$E = \frac{\int \dot{\Psi} H \Psi \partial r}{\int \dot{\Psi} \Psi \partial r}$$

Интегрировать надо по всем осям от $-\infty$ до $+\infty$.

Надо учитывать, что волновая может быть сложным числом и поэтому комплексная составляющая указывается явно.

```
#
print "$q $coor{$m} {$res} {"$N"}->x,"$n";
$nx=$nx+ $coor{$m} {$res} {"$N"}->x;
$ny=$ny+ $coor{$m} {$res} {"$N"}->y;
$nz=$nz+ $coor{$m} {$res} {"$N"}->z;

$ox=$ox+ $coor{$m} {$res} {"$O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"$O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"$O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use IO::File;
my $f = IO::File->new($ARGV[0], "r");
my $q = $f->readline();
```

Точное решение уравнения Шредингера

```
my ($my $coor, my $schnum)=read_pdb($ARGV[0]);
my $coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $schnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg } ;
```

```
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;
```

```
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/^.*\//";
$filename="-- s/\.pdb//";
$filename="-- s/\.dir/\.d";
system("cp $filename $dir/$filename");
print OUT "#INFO chain $schnum qnum $qnum \n";
```

```
foreach my $m (sort { $a<=>$b } keys %coor){
my %qartets = %qwa; #find_quart( $coor{$m} );
my %q = find_q( $coor{$m} );
```

```
# foreach my $q ( keys %qartets ){ print join " ", @{$qartets{$q}}, "\n";
```

```
foreach my $q ( keys %qartets ){
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
```

```
# print "$q $coor{$m} {$res} {"N"}->x, "\n";
$nx=$nx+ $coor{$m} {$res} {"N9"}->x;
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

Для решения уравнения надо интегрировать бесконечность, будем накладывать ограничения.

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Атомные единицы

```
my ($my %coor, my $schnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $schnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg } ;
```

```
my %qwa=find_quart( %coor{"0"} ); my $sqnum=keys %qwa;
```

```
my $filename=$schnum.".".$sqnum.".".$filename.".dat";
```

```
$filename="$mdir"."$filename".dat";
```

```
print "$filename";
```

```
open OUT, ">$filename";
```

```
print OUT "#INFO: chain $schnum $sqnum $sqnum\n";
```

```
foreach my $m ( sort { $a-<=>$b } keys %coor ){
```

```
my %q; foreach my $q ( keys %qartets ){
```

```
my %q; foreach my $q ( keys %qartets ){
```

```
# foreach my $q ( keys %qartets ){
```

```
foreach my $q ( keys %qartets ){
```

```
my $nx; my $ny; my $nz;
```

```
my $ox; my $oy; my $oz;
```

```
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
```

```
# print "$q $coor{$m}{$res} {"N9"}->x,"n";
```

```
$nx=$nx+ $coor{$m}{$res} {"N9"}->x;
```

```
$ny=$ny+ $coor{$m}{$res} {"N9"}->y;
```

```
$nz=$nz+ $coor{$m}{$res} {"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{$res} {"O6"}->x;
```

```
$oy=$oy+ $coor{$m}{$res} {"O6"}->y;
```

```
$oz=$oz+ $coor{$m}{$res} {"O6"}->z;
```

```
#
```

Работая с квантовой механикой мы оперируем частицами и если выразить для них энергию или массу в системе Си, то придётся ворочать значениями 10⁻¹⁰, упростили:

1 Au, mass

9.1093826(16) × 10⁻³¹

1 Au, charge

1.60217653(14) × 10⁻¹⁹

1 Hartree, energy

4.35974417(75) × 10⁻¹⁸

1 Bohr, length

5.291772108(18) × 10⁻¹¹

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Одно-электронный атом

$$H = -\frac{\hbar^2}{2m} \nabla^2 - \frac{Ze^2}{4\pi\epsilon_0 r} \text{ или в упрощенных единицах: } H = -\frac{1}{2} \nabla^2 - \frac{Z}{r}$$

Так как система имеет сферическую симметрию, то можно представить волновую функцию в сферических координатах.

```
if ($snum > 0) {
#system("mkdir SARGV[1]");
my $filename=SARGV[0];
$filename=$filename.".n";
$filename=$snum.".n";
$filename="sdir".$filename.".dat";
print "$filename\n";
open(FILE,">".$filename);

```

$$\psi_{nlm}(r, \vartheta, \varphi) = \sqrt{\left(\frac{2}{na_0}\right)^3 \frac{(n-l-1)!}{2n(n+l)!}} e^{-\rho/2} \rho^\ell L_{n-l-1}^{2\ell+1}(\rho) Y_\ell^m(\vartheta, \varphi);$$

$L_{n-l-1}^{2\ell+1}(\rho)$ - Обобщённый полином Лагерра степени $n-l-1$; $\rho = \frac{2r}{na_0}$

$Y_\ell^m(\vartheta, \varphi)$ - Сферическая гармоника;

Где n, l, m это основные квантовые числа

- ```
foreach my $q (keys %qartets) {
my $ix;
my $ox; my $oy; my $oz;
my $r;
foreach my $res (@qartets{$q}) {
#
my $ny=$ny+ $coor[$m]{$res}{"N9"}->x;
my $nz=$nz+ $coor[$m]{$res}{"N9"}->z;

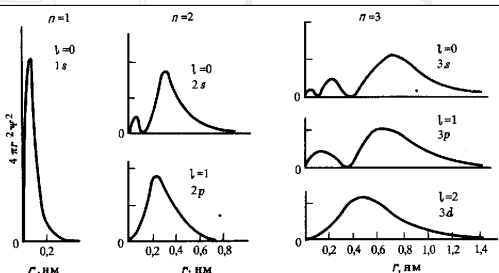
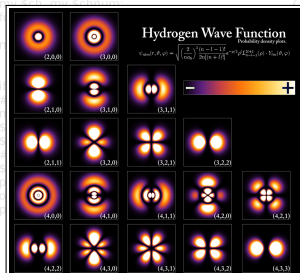
my $sx=$sx+ $coor[$m]{$res}{"O6"}->x;
my $sy=$sy+ $coor[$m]{$res}{"O6"}->y;
my $sz=$sz+ $coor[$m]{$res}{"O6"}->z;

```
- $n$  - основное число (1,2,3..)
  - $l$  - орбитальное число (0,1,2..  $n-1$ )
  - $m$  - магнитное число ( $-l..+l$ )

```
#!usr/bin/perl
use Math::VectorReal qw(:all);
```

# Одно-электронный атом

```
#[my %coor,my $chnum]=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
```



Тут есть красивая анимация которая на примере поверхности барабана объяснит откуда берутся такие формы.

[http://en.wikipedia.org/wiki/Atomic\\_orbital](http://en.wikipedia.org/wiki/Atomic_orbital)

```
foreach my $q (keys %quartets){
 my $ox; my $oy; my $oz;
 foreach my $res (@{ $quartets{$q} }) {
 print "$q $coor{$m} {$res} {"$N9"}->x,\"n\";
 $nx=$nx+ $coor{$m} {$res} {"$N9"}->x;
 $ny=$ny+ $coor{$m} {$res} {"$N9"}->y;
 $nz=$nz+ $coor{$m} {$res} {"$N9"}->z;
 }
 $ox=$ox+ $coor{$m} {$res} {"$O6"}->x;
 $oy=$oy+ $coor{$m} {$res} {"$O6"}->y;
 $oz=$oz+ $coor{$m} {$res} {"$O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Многоэлектронный атом

```
my ($coor,$snum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
```

Полное решение уравнение Шредингера для многоэлектронного атома затруднено, по ряду причин:

- N-body problem, суть вопроса, предсказать движение трёх и более тел на всём течении времени, если известны положение и скорости на текущий момент.
- Добавление четвёртого экспериментального квантового числа, спина, создаёт необходимость различать электроны.
- Квадрат волновой функции равен плотности. Трактование волновой функции как плотности электрона в данном месте, означает, что плотность может быть образована любым электроном. Этот факт сильно затрудняет расчёты.

```
if ($qnum) {
#system("perl $filename.pl $dir $snum");
my $filename=$ARGV[0];
$filename=$dir.$filename;
#system("perl $filename.pl $dir $snum");
print "$filename\n";
open OUT,">$filename";
print OUT "$snum\n";
}

foreach my $s ($snum) {
my %q=();
my %q= find_q(%coor($s));

for($i=0; $i<$snum; $i++) {
for($i=0; $i<$snum; $i++) {
my $sx=$snum-$i;
my $sy=$snum-$i;
my $sz=$snum-$i;

print "$q $coor($s) ($sres){\"N7\"}->x,\"n\";
$nx=$nx+ $coor($s) ($sres){\"N9\"}->x;
$ny=$ny+ $coor($s) ($sres){\"N9\"}->y;
$nz=$nz+ $coor($s) ($sres){\"N9\"}->z;

$ox=$ox+ $coor($s) ($sres){\"O6\"}->x;
$oy=$oy+ $coor($s) ($sres){\"O6\"}->y;
$oz=$oz+ $coor($s) ($sres){\"O6\"}->z;
}
}

}
```

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Приближение Борна-Оппергеймера

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $chnum;
foreach my $r (sort keys %{$coor{"O"}}) { my $qgg=substr($r,0,1); if ($qgg ne $sch) { $chnum++; $sch=$qgg; }
}

```

Если мы считаем, что ядра двигаются сильно медленнее чем электроны, то мы можем считать

$$\Psi_{total} = \psi_{electronic} \psi_{nucleic}$$

$$E_{total} = E_{electronic} + E_{nucleic}$$

Google:

electron mass =  $9.10938188 \times 10^{-31} \text{ kg}$

proton mass =  $1.67262158 \times 10^{-27} \text{ kg}$

# Атом гелия

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $idir=$ARGV[1];
my $sch, my $chnum;
foreach my $r (sort keys %{$coor{"0"}}){ my $ggg=subst("0,1",li($ggg,$sch))($chnum+1,$sch-$ggg);
```

$$(H_1 + H_2)\Psi(r_1, r_2) = E\Psi(r_1, r_2)$$

Спин орбиталь это:  $\chi_i(x_i) = \chi_i(r_i)\sigma(s_i)$

Давайте запишем волновую функцию электрона на самой низкой орбитали как:

$$\chi_1(x_1)\chi_2(x_2); \chi_1(x_2)\chi_2(x_1)$$

тогда:  $\Psi = \frac{1}{\sqrt{2}}[\chi_1(x_1)\chi_2(x_2) - \chi_1(x_2)\chi_2(x_1)]$

но  $\Psi = \frac{1}{\sqrt{2}}[\chi_1(x_1)\chi_1(x_2) - \chi_1(x_2)\chi_1(x_1)] = 0$

Вот он принцип Паули и матричное описание:

$$\Psi = \frac{1}{\sqrt{2}} \begin{vmatrix} \chi_1(x_1) & \chi_1(x_2) \\ \chi_1(x_1) & \chi_1(x_1) \end{vmatrix}$$



# Любая система

Итак для 1s можно построить матрицу:

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
my ($R,$N2,$N7,$R2,$R3) = @ARGV;
my ($my $coor,$my $schnum)=read_pdb($ARGV[0]);
my $my $coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch,$my $schnum;
foreach my $fr (sort keys %{$coor{"0"}}) {
 my $qwa=find_quart($coor{"0"});
 my $sqnum=
```

$$\Psi = \frac{1}{\sqrt{2}} \begin{vmatrix} \chi_1(r_1)\sigma_1(\alpha) & \chi_1(r_2)\sigma_2(\alpha) \\ \chi_1(r_1)\sigma_1(\beta) & \chi_1(r_2)\sigma_2(\beta) \end{vmatrix}$$

```
или >0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename-- s/^\.//;
$filename-- s/\.pdb//;
#$filename=$schnum."_$sqnum."_$filename;
$filename="$mdir".$filename.".dat";
print " $filename\n";
}
print OUT "chain schnum $schnum $sqnum\n";

foreach my $sm (sort {$a=<=>$b} keys %coor){
 my %qartets = %qwa; #find_quart($coor{$sm});
 my %q = find_q($coor{$sm});

foreach my $sq { keys %qartets } {
 foreach my $sq { keys %qartets } {
 my $nx; my $ny; my $nz;
 my $ox; my $oy; my $oz;
 my $r;

 foreach my $res (@{ $qartets{$sq} }) {
 print "$q $coor{$sm} {$res} {" $R }->x," $nx;
 $nx=$nx+ $coor{$sm} {$res} {" $N9 }->x;
 $ny=$ny+ $coor{$sm} {$res} {" $N9 }->y;
 $nz=$nz+ $coor{$sm} {$res} {" $N9 }->z;
 print " $R }->y," $ny;
 print " $R }->z," $nz;
 }
 }
}
```

$$= \frac{1}{\sqrt{2}} \chi_{1s}(r_1)\chi_{1s}(r_2) \begin{vmatrix} \sigma_1(\alpha) & \sigma_2(\alpha) \\ \sigma_1(\beta) & \sigma_2(\beta) \end{vmatrix}$$

Тогда мы переходим к N:

```
foreach my $sm (sort {$a=<=>$b} keys %coor){
 my %qartets = %qwa; #find_quart($coor{$sm});
 my %q = find_q($coor{$sm});

foreach my $sq { keys %qartets } {
 foreach my $sq { keys %qartets } {
 my $nx; my $ny; my $nz;
 my $ox; my $oy; my $oz;
 my $r;

 foreach my $res (@{ $qartets{$sq} }) {
 print "$q $coor{$sm} {$res} {" $R }->x," $nx;
 $nx=$nx+ $coor{$sm} {$res} {" $N9 }->x;
 $ny=$ny+ $coor{$sm} {$res} {" $N9 }->y;
 $nz=$nz+ $coor{$sm} {$res} {" $N9 }->z;
 print " $R }->y," $ny;
 print " $R }->z," $nz;
 }
 }
}
```

$$\Psi = \frac{1}{\sqrt{N}} \begin{pmatrix} \chi_1(1) & \chi_1(2) & \cdots & \chi_1(N) \\ \chi_2(1) & \chi_2(2) & \cdots & \chi_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_N(1) & \chi_N(2) & \cdots & \chi_N(N) \end{pmatrix}$$

это и есть **Определитель Слейтора**, орбитальное приближение волновой функции.

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Расчёт энергии для молекулы водорода

```
my ($my $coor, my $chnum)=read_pdb($ARGV[0]);
```

```
my %coor=read_pdb($ARGV[0]);
```

Воспользуемся теорией молекулярных орбиталей и будем считать, что каждая молекулярная орбиталь есть линейная комбинация атомных орбиталей:

```
if ($qnum > 0){
```

```
 #system("mkdir $ARGV[1]");
```

```
 my $filename=$ARGV[0];
```

```
 $filename-- s/\^\.//;
```

```
 $filename-- s/\.pdb//;
```

```
 # $filename=$chnum.".".$qnum.".".$filename.".dat";
```

```
 $filename="dir".$filename.".dat";
```

```
 print "$filename\n";
```

```
 open OUT,">$filename";
```

```
 print OUT.">INFO chain $chnum atom $qnum\n";
```

```
 my %qartets = %qwa; #find quartl $coor{$sm};
```

Линейная комбинация двух 1s орбиталей:  $1\sigma_g = A(1s_A + 1s_B)$ , где A это коэффициент нормализации.

```
foreach my $q { keys %qartets } { print join " ", @{$qartets{$q}}, "\n";
```

```
 foreach my $q { keys %qartets{
```

```
 my $nx; my $ny; my $nz;
```

```
 my $ox; my $oy; my $oz;
```

```
 my $r;
```

```
 foreach my $res (@{$qartets{$q}}){
```

```
 $x=$nx+ $coor{$sm}{ $res }{"N9"}->x;
```

```
 $ny=$ny+ $coor{$sm}{ $res }{"N9"}->y;
```

```
 $nz=$nz+ $coor{$sm}{ $res }{"N9"}->z;
```

```
 $ox=$ox+ $coor{$sm}{ $res }{"O6"}->x;
```

```
 $oy=$oy+ $coor{$sm}{ $res }{"O6"}->y;
```

```
 $oz=$oz+ $coor{$sm}{ $res }{"O6"}->z;
```

$$\psi_i = \sum_{\mu=1} K c_{\mu i} \phi_{\mu}$$

$$\Psi = \begin{vmatrix} \chi_1(1) & \chi_2(1) \\ \chi_2(1) & \chi_2(2) \end{vmatrix}$$

Где  $\chi_1(1) = 1\sigma_g(1)\alpha(1)$  и т.д.

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Расчёт энергии для молекулы водорода

Подставляем матрицу полученную выше в гамильтониан:

$$H = -\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{Z_A}{r_{1A}} - \frac{Z_B}{r_{1B}} - \frac{Z_A}{r_{2B}} - \frac{Z_B}{r_{2A}} + \frac{1}{r_{12}}$$

```
if ($qnum > 0){
#system("mkdir SARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/~/V//;
$filename="-- s/./pdb//;
#$filename=$chnum."."$qnum."."$filename.".dat";
$filename="$dir"."$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";
```

$$E = \frac{\int \Psi H \Psi \partial r}{\int \Psi \Psi \partial r}$$

```
foreach my $m (sort {$a<=>$b} keys %coor){
my %qartets = %qwa; #find quart($coor{$m});
my %q = find_q($coor{$m});
```

$$E = 1/2 \int \int \partial \tau_1 \tau_2 ([\chi_1(1)\chi_2(2) - \chi_2(1)\chi_1(2)])$$

$$\left[ -1/2\nabla_1^2 - 1/2\nabla_2^2 - \frac{1}{r_{1A}} - \frac{1}{r_{1B}} - \frac{1}{r_{2A}} - \frac{1}{r_{2B}} + \frac{1}{r_{12}} \right]$$

$$[\chi_1(1)\chi_2(2) - \chi_2(1)\chi_1(2)]$$

```
$ox=$ox+ $coor{$m}{ $res{"O6"}->x;
$oy=$oy+ $coor{$m}{ $res{"O6"}->y;
$oz=$oz+ $coor{$m}{ $res{"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Расчёт энергии в общем виде

```

#(my %coor,my $chnum)=read_pdb($RGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $q = ($ARGV[2], $ARGV[3]);
foreach my $r (sort keys %{$coor}) {
 my $ggg=$coor{$r}[0]; if ($ggg eq "O") { $r=$r+$sch; }
 my $r12 = sqrt($coor{$r}[0]**2 + $coor{$r}[1]**2 + $coor{$r}[2]**2);
 my $r13 = sqrt($coor{$r}[0]**2 + $coor{$r}[1]**2 + $coor{$r}[3]**2);
}

```

$$H = \left( -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \frac{1}{2} \nabla_2^2 - \frac{Z_A}{r_{1A}} - \frac{Z_B}{r_{1B}} \cdots + \frac{1}{r_{12}} + \frac{1}{r_{13}} + \dots \right)$$

```
my %qwa=find_quart($coor{"O"}); my $qnum=keys %qwa;
```

## Полный гамильтониан и матрица Слейтора:

```

if ($qnum > 0) {
 #system("mkdir $ARGV[1]");
 my $filename=$ARGV[0];
 $filename="-- s/^.*\//";
 $filename="-- s/\.pdb//";
 # $filename=$chnum.".".$qnum.".".$filename.".dat";
 $filename="$dir"."$filename.".dat";
 print "$filename\n";
 open OUT,">$filename";
 print OUT "#INFO chain $chnum qnum $qnum\n";
}

```

$$\Psi = \frac{1}{\sqrt{N!}} \begin{pmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_N(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(N) & \chi_2(N) & \cdots & \chi_N(N) \end{pmatrix}$$

```

foreach my $m (sort {$a=<=>$b} keys %coor) {
 my %qartets = %qwa; #find_quart($coor{$m});
 my %q = find_q($coor{$m});
}

```

```
foreach my $q (keys %qartets) { print join " ", @qartets{$q} }, "\n";
```

$$\int \Psi H \Psi = \int \dots \int \partial \tau_1 \tau_2 \dots \tau_N \{ [\chi_i(1) \chi_j(2) \chi_k(3) \dots]$$

```

foreach my $res (@qartets{$q}) {
 print "$q $coor{$res}[0] $coor{$res}[1] $coor{$res}[2] $coor{$res}[3]";
 $nx=$nx+$coor{$res}[0]; $res{"O6"}->x;
 $ny=$ny+$coor{$res}[1]; $res{"O6"}->y;
 $nz=$nz+$coor{$res}[2]; $res{"N9"}->z;
}

```

$$\left( \frac{1}{2} \sum_i \nabla_i^2 - 1/r_{1A} - 1/r_{1B} \cdots + 1/r_{12} + 1/r_{13} + \dots \right)$$

```

$ox=$ox+$coor{$m}[$res>{"O6"}->x;
$oy=$oy+$coor{$m}[$res>{"O6"}->y;
$oz=$oz+$coor{$m}[$res>{"O6"}->z;
}

```

$$([\chi_i(1) \chi_j(2) \chi_k(3) \dots])$$

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Разделим на компоненты

```

#(my %$coor,$my $schnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $idir=$ARGV[1];
my $sch,$my $schnum;
foreach my $r (sort keys %{$coor{"0"}}) { my $ggg=(substr($r,0,1); if ($ggg ne "$sch") { $sch=$ggg; $schnum++; } } }

```

$$E_{ij}^{core} = \int \partial\tau_1 \chi_i(1) \left( -\frac{1}{2} \nabla_i^2 - \sum_{A=1}^M \frac{Z_A}{r_{iA}} \right) \chi_j(1)$$

```
my %$qwa=find_quart($coor{"0"}); my $sqnum=keys %$qwa;
```

Для мультиэлектронной системы кинетическая и потенциальная энергия:

```

my $idir=$ARGV[1];
$filename="-- s/^.*\//";
$filename="-- s/\.pdb//";
#$filename="$schnum"."$sqnum"."$filename.".dat";
$filename="$idir"."$filename".dat;
print "$filename\n";
open OUT,">$filename";
print OUT,"#INFO chain $schnum $sqnum\n";

```

$$E_{total}^{core} = \sum_{i=1}^N \int \partial\tau_1 \chi_i(1) \left( -\frac{1}{2} \nabla_i^2 - \sum_{A=1}^M \frac{Z_A}{r_{iA}} \right) \chi_j(1) = \sum_{i=1}^N H_{ij}^{core}$$

Электростатическое отталкивание электронов:

```
foreach my $q (keys %$qartets) { print join " ", @{$qartets{$q}}, "\n"; }
```

$$E_{ij}^{Coulomb} = \int \int \partial\tau_1 \partial\tau_2 \chi_i(1) \chi_j(1) \frac{1}{r_{12}} \chi_i(2) \chi_j(2)$$

$$E_{total}^{Coulomb} = \sum_{i=1}^N \sum_{j=i+1}^N \int \int \partial\tau_1 \partial\tau_2 \chi_i(1) \chi_j(1) \frac{1}{r_{12}} \chi_i(2) \chi_j(2) = \sum_{i=1}^N \sum_{j=i+1}^N J_{ij}$$

```

$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;

```

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Особенность квантовой природы электрона

```
my ($coor,$snum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
```

Два электрона с одинаковыми спинами в рамках принятой модели имеют поправку к электростатическому отталкиванию, делая его менее значимым:

```
if ($snum > 0){
```

```
 #system("mkdir $ARGV[1]");
 my $filename=$ARGV[0];
```

```
 $filename-- s/^\.//;
 $filename-- s/\.pdb//;
```

```
 # $filename=$snum.".".$snum.".7.$filename.".$snum.".dat";
 $filename="dir".$filename.".dat";
```

```
 print "$filename\n";
 open OUT,">$filename";
```

```
 print OUT "#INFO chain $snum group $snum\n";
```

```
 foreach my $m (sort {$s{$snum}{$b} keys %coor}
 my %qartets = %qwa; #find quart ($coor{$m})
 my %q = find_q($coor{$m});
```

```
 # foreach my $q { keys %qartets } { print join " ", @qartets{$q}, "\n";
```

```
 foreach my $q { keys %qartets } {
 my $nx, my $ny, my $nz;
 my $ox, my $oy, my $oz;
 my $r;
```

Интеграл  $K_{ij}$  не ноль, только при одинаковом спине электронов.

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

# Упрощенные записи

```
#!/(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r (sort keys %{$coor{"O"}}){ my $ggg=substr($r,0,1); if ($ggg ne $ch){ $chnum++; $ch=$ggg };
```

Кулоновское отталкивание:

```
my %qwa=find_quart(%coor{"O"}); my $qnum=keys %qwa;
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename-- s/^.*\//;
$filename-- s/\.pdb//;
#$filename=$chnum.".".$qnum.".".$filename.".dat";
$filename="$dir".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
```

$$J_{ij} = \langle \chi_i \chi_j | \frac{1}{r_{12}} | \chi_i \chi_j \rangle$$

Забавно, что exchange интегралы сокращают до:

```
foreach my $m (sort {$a<=>$b} keys %coor){
my %qartets = %qwa; #find_quart(%coor{$m});
my %q = find_q(%coor{$m});
```

```
foreach my $q (keys %qartets) { print join " ", @
```

```
foreach my $q (keys %qartets){
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res (@ { $qartets{$q} }){
```

```
print "$q %coor{$m} {$res} {"R"}->x,"n";
```

```
$nx=$nx+ %coor{$m} {$res} {"N9"}->x;
```

```
$ny=$ny+ %coor{$m} {$res} {"N9"}->y;
```

```
$nz=$nz+ %coor{$m} {$res} {"N9"}->z;
```

```
$ox=$ox+ %coor{$m} {$res} {"O6"}->x;
```

```
$oy=$oy+ %coor{$m} {$res} {"O6"}->y;
```

```
$oz=$oz+ %coor{$m} {$res} {"O6"}->z;
```

$$K_{ij} = \langle \chi_i \chi_j | \frac{1}{r_{12}} | \chi_j \chi_i \rangle$$

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
```

## Итак, энергия:

Для не возбуждённых состояний или closed shell:

```
#!/my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r (sort keys %{$coor{"0"}}){ my $ggg=substr($r,0,1); my $nnum++; $ch=$ggg };
my %qwa=find_quart($coor{"0"}); my $qnum=keys %qwa;
```

$$\sum_{i=1}^{N/2} 2H_{ii}^{core}$$

Существует четыре способа: как электроны с одной орбитали взаимодействуют с электронами с другой орбитали и есть всего два способа получить спаренные электроны и:

```
foreach my $m (sort {$a<=>$b} keys %qartets){
 my %qartets = %qwa; #find_quart($coor{$m});
 my %q = find_q($coor{$m});
```

$$E = 2 \sum_{i=1}^{N/2} 2H_{ii}^{core} + \sum_{i=1}^{N/2} \sum_{j=i+1}^{N/2} (4J_{ij} - 2K_{ij}) + \sum_{i=1}^{N/2} J_{ii}$$

Или если примем  $J_{ii} = K_{ii}$

```
foreach my $res (@{ $qartets{$q} }){
 print "$q $coor{$m} {$res} {"R";
 $nx=$nx+ $coor{$m} {$res} {"N9"}->x;
 $ny=$ny+ $coor{$m} {$res} {"N9"}->y;
 $nz=$nz+ $coor{$m} {$res} {"N9"}->z;
 $ox=$ox+ $coor{$m} {$res} {"O6"}->x;
 $oy=$oy+ $coor{$m} {$res} {"O6"}->y;
 $oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

$$E = 2 \sum_{i=1}^{N/2} 2H_{ii}^{core} + \sum_{i=1}^{N/2} \sum_{j=i+1}^{N/2} (J_{ij} - K_{ij})$$