

ВАРИАТОР – УПРАВЛЕНИЕ ВАРИАНТАМИ ВЫЧИСЛЕНИЙ

В большинстве реальных расчетов профилей линий требуется рассчитать сразу много случаев. Иногда нужно варьировать один параметр, чтобы найти соответствие расчетного профиля экспериментальному, но обычно нужно задавать вариации нескольких параметров. Если результаты сравнивать в виде графиков профилей Стокса, то графики можно расположить

- в ряд,
- в колонку,
- можно менять цвет, стиль графиков,
- можно делать мультипликацию картинки (если просмотр делается на экране).

Реально имеет смысл в одном расчете изменять два – три – четыре параметра. Но при расчете обычно приходится задавать значительно больше исходных параметров, более десятка.

Таким образом, структура для управления параметрами должна содержать **список параметров**, из которых первые несколько – многомерные, остальные включают по одному значению.

Количество посчитанных профилей в одном расчете равно произведению размерностей всех параметров.

Номера вариантов расчета составляют многомерную матрицу. Для наглядности обратимся к примеру расчета профилей Стокса по аналитическим формулам Унно – Рачковского. Результаты зависят от ряда параметров, в первую очередь от

VH	- магнитного уширения
GM	- угла между вектором поля и лучом зрения
ETA0	- отношения коэффициента поглощения в центре линии и в континууме и
A	- параметра, определяющего лорнцевскую часть профиля Фойгта.

Предположим мы хотим посчитать профили для линии типа Fe I 6302 A для

VH	= 0.5,	1.0,	1.5,	2.0			
GM	= 0,	15,	30,	45,	60,	75,	90 градусов
ETA0	= 20,	50,	100,	200			
A	= 0.002,	0.02,	0.2				

У нас получается соответственно 4,7,4 и 3 значения – всего $4*7*4*3 = 336$ комбинаций. Для профилей нет смысла заводить 4-х мерный массив, т.к. в следующей задаче размерность может быть и 2 и 3 и 5, их лучше хранить в линейном массиве или списке размером 336 элементов. Объект VARIATOR же должен иметь в нашем случае 4 записи. Для каждой записи должны быть заданы – имя параметра, число вариантов и массив или список значений вариантов. Сам VARIATOR должен знать какая у него размерность – сколько у него варьируемых переменных. Таким образом, получаем структуру

```

type
  TVals = class(TList)  (* список значений переменных *)
    (* в списке TVals хранятся значения параметров *)
    (* для начала считаем, что они имеют тип REAL *)
  end;

  TVariRec = record (* описывает один физический параметр*)
    Key : string; (* имя параметра *)
    Vals : TVals; (* список значений параметра *)
    (* Vals.Count - число значений данного параметра *)
  end;

  TVARIATOR = class(TList) (* список физических параметров *)
    (* в списке TVARIATOR хранятся записи TVariRec *)
    (* Self.Count - размерность массива вариантов *)
    NK : integer; (* общее число комбинаций *)
    Kidx : integer; (* текущий индекс - от 1 до NK *)
  end;

```

Добавление параметра в VARIATOR приводит к увеличению его размерности и выглядит примерно так, вызов:

```
VARIATOR.AddP('#VH 0.5 1.0 1.5 2.0');
VARIATOR.AddP('#GM 0 15 30 45 60 75 90');
VARIATOR.AddP('#ETA0 20 50 100 200');
VARIATOR.AddP('#A 0.002 0.02 0.2');
```

описание процедуры:

```
procedure TVARIATOR.AddP(S:string);
type P_real = ^real; (* указатель, ссылка на значение типа real *)
var V : TVariRec;
    N : integer; (* число слов в строке S *)
    I : integer; (* текущий индекс слова в строке S *)
    W : string; (* копия текущего слова *)
    R : P_real; (* место, для копии текущего значения *)
begin
  N := GetNWord(S); (* кол-во слов, разделенных пробелами в строке S *)
  V.Key := GetWordN(S,1); (* первое слово - ключ, имя параметра *)
  V.Vals := TVals.Create; (* отведем место для списка параметров *)
  for I := 2 to N do begin
    W := GetWordN(S,I); (* получим слово из строки *)
    GetMem(R,SizeOf(real)); (* отведем место, чтобы хранить значение *)
    R^ := StringToReal(W); (* преобразуем подстроку W в число типа real *)
    V.Vals.Add(pointer(R)); (* добавим к списку значений *)
  end;
  RecalcNK; (* посчитаем заново общее число комбинаций *)
end; (* AddP *)
```

Когда значения параметров в VARIATOR загружены, он должен быть использован для перебора всех значений для любого из заданных параметров и, наоборот, для выдачи совокупности параметров по заданному общему номеру комбинации.

Так для нашего примера должны работать функции:

```
NI := VARIATOR.GetPNI('#VH') =>NI= 4 (* размерность параметра *)
J := VARIATOR.GetPJ('#ETA0')=>J = 3 (* номер параметра (считая от единицы) *)
R := VARIATOR.GetPIR('#A',2)>R = 0.02 (* второе значение параметра #A *)
S := VARIATOR.GetKS(125) =>S = '#VH=1.0 #GM=45 #ETA0=50 #A=0.02'
```

Для более сложных случаев, чем расчет профилей Стокса по формулам Унно – Рачковского входные параметры могут быть не только числами.

Это могут быть

- опции учитывать / не учитывать аномальную дисперсию
- целые числа количество точек на профиле
- таблицы распределение температуры с оптической глубиной
- другие структуры список атомов для учета среднего атомного веса,
 система солнечных сил осцилляторов определенных авторов

Произвольные, не только типа real параметры.

Как универсальное решение можно предложить хранить значение параметра в виде текстовой строки. Строка может содержать в себе не только собственно строку, но и ссылку на более сложные данные – например, имя файла с таблицей модели атмосферы. Включение такой возможности изменит VARIATOR следующим образом:

```
TVals = class(TStringList) (* список значений переменных *)
. . .
procedure TVARIATOR.AddP(SL:TStringList);
var V : TVariRec;
    I : integer; (* текущий индекс строки в списке строк SL *)
    W : string; (* копия текущей подстроки *)
```

```

begin
  N := SL.Count;          (* размер списка строк *)
  V.Key := SL.Strings[0]; (* первая подстрока - ключ, имя параметра *)
  V.Vals := TVals.Create; (* отведем место для списка значений параметра *)
  for I := 1 to SL.Count-1 do begin
    W := SL.Strings[I];  (* получим очередную подстроку *)
    V.Vals.Add(W);       (* добавим к списку строк описаний параметра *)
  end;
  RecalcNK;              (* посчитаем заново общее число комбинаций *)
end; (* AddP *)

```

Варианты, как комбинация нескольких параметров.

Предположим, нам нужно рассчитать набор профилей для мелкомасштабных фотосферных элементов с магнитным полем для разных положений на диске Солнца. Если мы предполагаем, что магнитное поле в этих элементах направлено вертикально, то меняя $\mu = \cos \theta$ - косинус гелиоцентрического угла мы должны **синхронно** менять угол наклона силовых линий к лучу зрения. Таким образом, одну варианту расчета соответствует изменение не одного параметра, а набора из двух и более параметров.

Этого можно достичь, если добавить к индексу записи TVaryRec добавить индекс варианта в вариаторе:

```

TVariRec = record
  Key   : string;      (* имя параметра *)
  Vals  : TVals;      (* список строковых описаний параметра *)
  JNum  : integer;    (* индекс варианта в вариаторе *)
end;

```

Физические параметры с одинаковым номером JNum будут изменяться синхронно при переборе вариантов расчета. Для добавления таких параметров к вариатору используется процедура ConcateP вместо AddP. Вызывать ConcateP нужно сразу же после AddP того параметра, к которому мы подсоединяем синхронный. Число вариантов ведущего и подсоединяемых параметров должно совпадать.

О терминах и обозначениях вариатора.

Сделаем определения.

Параметр	– физический или управляющий вычислениями параметр. Например, косинус гелиоцентрического угла, опция пренебрежения аномальной дисперсией или целая таблично заданная модель атмосферы.
Значение параметра	– данные, которые используются в единичном расчете.
Вариант	– совокупность параметров, значения которых синхронно меняются
Комбинация	– набор индексов каждого параметра I для всех вариантов L

Индекс значений	– I	- индекс варианта одного физ. параметра
Индекс параметра	– J	- индекс параметра в списке физ. параметров
Индекс варианта	– L	- индекс варианта расчета в массиве вариатора
Индекс комбинации	– K	- глобальный индекс варианта расчета

NI – Число значений одного варианта (или одного параметра номер J)

MJ – Число физических параметров всего, в том числе -

NJ – число параметров, имеющих более одного значения,

ML - размерность вариатора, число независимых координат для вариаций

NK – Общее число комбинаций