







## Как работи?

1. **Въведете** текста и формулите в прозореца "Код" отляво.
2. Натиснете **F5** или бутона  за да **изчислите** резултатите. Те ще се покажат отдясно в прозореца "Резултати", като професионално оформена изчислителна **записка**.
3. Натиснете  за да **отпечатате** или  за да **копирате** текста на записката.  
Може също да го **запишете** като **Html** , **PDF**  или **MS Word**  документ.

## Програмен език

Програмният език на **Calcpad** включва следните елементи (кликнете за вмъкване):

- Реални числа: цифри "0" - "9" и десетична точка ".";
- Комплексни числа:  $re \pm imi$  (например  $3 - 2i$ );
- Реални вектори:  $[v_1; v_2; v_3; \dots; v_n]$ ;
- Реални матрици:  $[M_{11}; M_{12}; \dots; M_{1n} \mid M_{21}; M_{22}; \dots; M_{2n} \dots \mid M_{m1}; M_{m2}; \dots; M_{mn}]$ ;
- Променливи:
  - Латински букви:  $a - z, A - Z$ ;
  - Гръцки букви:  $\alpha - \omega, A - \Omega$ ;
  - цифри:  $0 - 9$ ;
  - запетая: ",";
  - прим(ове): ' , ' , ' , ' , ' ;
  - горни индекси:  $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, n, +, -$ ;
  - специални символи:  $\bar{\phantom{x}}, \emptyset, \varnothing, ^\circ, \pounds$ ;
  - "\_" за долен индекс;

Имената на променливи трябва да започват с буква. Различава главни и малки букви.

- Константи:  
 $\pi, e, \varphi, \gamma, g, G, M_E, M_S, c, h, \mu_0, \varepsilon_0, k_e, e, m_e, m_p, m_n, N_A, \sigma, k_B, R, F, \gamma_c, \gamma_s, \gamma_a, \gamma_g, \gamma_w$
- Оператори:
  - "!" - факториел;
  - "^" - степенуване;
  - "/" - делене;
  - "÷" - дробна черта;
  - "\" - целочислено делене;
  - "⊗" - остатък;
  - "\*" - умножение;
  - "-" - изваждане;
  - "+" - събиране;
  - "=" - равенство;
  - "≠" - неравенство;
  - "<" - по-малко;
  - ">" - по-голямо;
  - "≤" - по-малко или равно;
  - "≥" - по-голямо или равно;

" $\wedge$ " - логическо "И" (AND);

" $\vee$ " - логическо "ИЛИ" (OR);

" $\oplus$ " - изключващо "ИЛИ" (XOR);

"=" - присвояване;

- Потребителски функции от вида  $f(x; y; z; \dots)$ ;

- Вградени функции:

Тригонометрични:

$\sin(x)$  - синус;

$\cos(x)$  - косинус;

$\tan(x)$  - тангенс;

$\csc(x)$  - косеканс;

$\sec(x)$  - секанс;

$\cot(x)$  - котангенс;

Хиперболични:

$\sinh(x)$  - синус хиперболичен;

$\cosh(x)$  - косинус хиперболичен;

$\tanh(x)$  - тангенс хиперболичен;

$\csch(x)$  - косеканс;

$\sech(x)$  - секанс;

Обратни тригонометрични:

$\coth(x)$  - котангенс хиперболичен;

$\asin(x)$  - аркуссинус;

$\acos(x)$  - аркускосинус;

$\atan(x)$  - аркустангенс;

$\atan2(x; y)$  - ъгъл, чиито тангенс е отношението на  $y$  към  $x$ ;

$\acsc(x)$  - аркускосеканс;

$\asec(x)$  - аркуссеканс;

$\acot(x)$  - аркускотангенс;

Обратни хиперболични:

$\asinh(x)$  - аркуссинус хиперболичен;

$\acosh(x)$  - аркускосинус хиперболичен;

$\atanh(x)$  - аркустангенс хиперболичен;

$\acsch(x)$  - аркускосеканс хиперболичен;

$\asech(x)$  - аркуссеканс хиперболичен;

$\acoth(x)$  - аркускотангенс хиперболичен;

Логаритмични, експоненциални и корени:

$\log(x)$  - десетичен логаритъм;

$\ln(x)$  - натурален логаритъм;

$\log_2(x)$  - двоичен логаритъм;

$\exp(x)$  - експоненциална функция;

**sqr**(*x*) или **sqrt**(*x*) - корен квадратен;

**cbrt**(*x*) - корен кубичен;

**root**(*x*; *n*) - корен n-ти;

Закръгляване:

**round**(*x*) - закръгляване до най-близкото цяло число;

**floor**(*x*) - закръгляване до по-малкото цяло число;

**ceiling**(*x*) - закръгляване до по-голямото цяло число;

**trunc**(*x*) - закръгляване към по-близкото число в посока към нулата;

Целочислени:

**mod**(*x*; *y*) - остатък от деление;

**gcd**(*x*; *y*; *z*...) - най-голям общ делител;

**lcm**(*x*; *y*; *z*...) - най-малко общо кратно;

Комплексни:

**abs**(*x*) - абсолютна стойност;

**re**(*x*) - реалната част на комплексно число;

**im**(*x*) - имагинерната част на комплексно число;

**phase**(*x*) - фаза на комплексно число;

Агрегатни и интерполационни:

**min**(*x*; *y*; *z*...) - минимум на множество стойности;

**max**(*x*; *y*; *z*...) - максимум на множество стойности;

**sum**(*x*; *y*; *z*...) - сума на множество стойности =  $x + y + z \dots$ ;

**sumsq**(*x*; *y*; *z*...) - сума от квадратите =  $x^2 + y^2 + z^2 \dots$ ;

**srss**(*x*; *y*; *z*...) - корен квадратен от сумата на квадратите = **sqrt**( $x^2 + y^2 + z^2 \dots$ );

**average**(*x*; *y*; *z*...) - средно аритметично от множество стойности =  $(x + y + z \dots) / n$ ;

**product**(*x*; *y*; *z*...) - произведение на множество стойности =  $x \cdot y \cdot z \dots$ ;

**mean**(*x*; *y*; *z*...) - средно геометрично = n-th **root**( $x \cdot y \cdot z \dots$ );

**take**(*x*; *a*; *b*; *c*; ...) - връща n-тия елемент от списъка;

**line**(*x*; *a*; *b*; *c*; ...) - линейна интерполация;

**spline**(*x*; *a*; *b*; *c*; ...) - spline интерполация на Ермит;

Условни и логически:

**if**(*условие*; *стойност-при-истина*; *стойност-при-неистина*) - условно изчисление;

**switch**(*усл1*; *стойност1*; *усл2*; *стойност2*; ... ; *стойност-по-подразб.*) - избиращо

изчисление;

**not**(*x*) - логическо отрицание (NOT);

**and**(*x*; *y*; *z*...) - логическо "И" (AND);

**or**(*x*; *y*; *z*...) - логическо "ИЛИ" (OR);

**xor**(*x*; *y*; *z*...) - изключващо "ИЛИ" (XOR);

Други:

**sign**(*x*) - знак на число;

**random**(*x*) - произволно число между 0 и *x*;

## Векторни:

### Създаване и инициализация:

- vector**( $n$ ) - създава празен вектор с дължина  $n$ ;
- fill**( $\vec{v}; x$ ) - запълва вектора  $\vec{v}$  със стойност  $x$ ;
- range**( $x_1; x_n; s$ ) - създава вектор от стойностите в интервала от  $x_1$  до  $x_n$  със стъпка  $s$ ;

### Структурни:

- len**( $\vec{v}$ ) - връща дължината на вектора  $\vec{v}$ ;
- size**( $\vec{v}$ ) - действителния размер на вектора  $\vec{v}$  (индекса на последния ненулев елемент);
- resize**( $\vec{v}; n$ ) - задава нова дължина  $n$  на вектора  $\vec{v}$ ;
- join**( $A; \vec{b}; c...$ ) - създава вектор чрез обединяване на аргументите в списъка - матрици, вектори и скалари;
- slice**( $\vec{v}; i_1; i_2$ ) - връща частта от вектора  $\vec{v}$ , ограничена от индекси  $i_1$  и  $i_2$ , включително;
- first**( $\vec{v}; n$ ) - първите  $n$  елемента на вектора  $\vec{v}$ ;
- last**( $\vec{v}; n$ ) - последните  $n$  елемента на вектора  $\vec{v}$ ;
- extract**( $\vec{v}; \vec{i}$ ) - извлича онези елементи от  $\vec{v}$ , чиито индекси се съдържат в  $\vec{i}$ ;

### Данни:

- sort**( $\vec{v}$ ) - сортира вектора  $\vec{v}$  във възходящ ред;
- rsort**( $\vec{v}$ ) - сортира вектора  $\vec{v}$  в низходящ ред;
- order**( $\vec{v}$ ) - индексите на  $\vec{v}$ , подредени по възходящия ред на неговите елементи;
- revorder**( $\vec{v}$ ) - индексите на  $\vec{v}$ , подредени по низходящия ред на неговите елементи;
- reverse**( $\vec{v}$ ) - нов вектор, съдържащ елементите на  $\vec{v}$  в обратен ред;
- count**( $\vec{v}; x; i$ ) - броя на елементите в  $\vec{v}$ , от  $i$ -тия нататък, които са равни на  $x$ ;
- search**( $\vec{v}; x; i$ ) - индекса на първия елемент в  $\vec{v}$ , от  $i$ -тия нататък, който е равен на  $x$ ;
- find**( $\vec{v}; x; i$ ) или
- find\_eq**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $= x$ ;
- find\_ne**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $\neq x$ ;
- find\_lt**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $< x$ ;
- find\_le**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $\leq x$ ;
- find\_gt**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $> x$ ;
- find\_ge**( $\vec{v}; x; i$ ) - индексите на всички елементи в  $\vec{v}$ , от  $i$ -тия нататък, които са  $\geq x$ ;
- lookup**( $\vec{a}; \vec{b}; x$ ) или
- lookup\_eq**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $= x$ ;
- lookup\_ne**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $\neq x$ ;

**lookup\_lt**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $< x$ ;  
**lookup\_le**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $\leq x$ ;  
**lookup\_gt**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $> x$ ;  
**lookup\_ge**( $\vec{a}; \vec{b}; x$ ) - всички елементи в  $\vec{a}$ , за които съответните елементи в  $\vec{b}$  са  $\geq x$ ;

#### Математически:

**norm\_1**( $\vec{v}$ ) - L1 (Манхатън) норма на вектора  $\vec{v}$ ;  
**norm**( $\vec{v}$ ) или  
**norm\_2**( $\vec{v}$ ) или  
**norm\_e**( $\vec{v}$ ) - L2 (Евклидова) норма на вектора  $\vec{v}$ ;  
**norm\_p**( $\vec{v}; p$ ) - Lp норма на вектора  $\vec{v}$ ;  
**norm\_i**( $\vec{v}$ ) -  $L^\infty$  (безкрайна) норма на вектора  $\vec{v}$ ;  
**unit**( $\vec{v}$ ) - нормализирания (единичен) вектор  $\vec{v}$  (с L2 норма = 1);  
**dot**( $\vec{a}; \vec{b}$ ) - скалярно произведение на два вектора  $\vec{a}$  и  $\vec{b}$ ;  
**cross**( $\vec{a}; \vec{b}$ ) - векторно произведение на два вектора  $\vec{a}$  и  $\vec{b}$  (с дължина 2 или 3);

#### Матрични:

##### Създаване и инициализация:

**matrix**( $m; n$ ) - създава празна матрица с размери  $m \times n$ ;  
**identity**( $n$ ) - създава единична матрица с размери  $n \times n$ ;  
**diagonal**( $n; d$ ) - създава диагонална  $n \times n$  матрица и запълва главния диагонал със стойност  $d$ ;  
**column**( $m; c$ ) - създава матрица-стълб с размери  $m \times 1$ , запълнена със стойност  $c$ ;  
**utriang**( $n$ ) - създава горна триъгълна матрица с размери  $n \times n$ ;  
**ltriang**( $n$ ) - създава долна триъгълна матрица с размери  $n \times n$ ;  
**symmetric**( $n$ ) - създава симетрична матрица с размери  $n \times n$ ;  
**vec2diag**( $\vec{v}$ ) - създава диагонална матрица от елементите на вектора  $\vec{v}$ ;  
**vec2col**( $\vec{v}$ ) - създава матрица-стълб от елементите на вектора  $\vec{v}$ ;  
**join\_cols**( $\vec{c}_1; \vec{c}_2; \vec{c}_3 \dots$ ) - създава нова матрица чрез обединяване на вектори в стълбове;  
**join\_rows**( $\vec{r}_1; \vec{r}_2; \vec{r}_3 \dots$ ) - създава нова матрица чрез обединяване на вектори в редове;  
**augment**( $A; B; C \dots$ ) - създава нова матрица чрез присъединяване на матриците  $A, B, C \dots$  една до друга;  
**stack**( $A; B; C \dots$ ) - създава нова матрица чрез присъединяване на матриците  $A, B, C \dots$  една под друга;

##### Структурни:

**n\_rows**( $M$ ) - броя на редовете в матрицата  $M$ ;  
**n\_cols**( $M$ ) - броя на стълбовете в матрицата  $M$ ;  
**resize**( $M; m; n$ ) - задава нови размери  $m$  и  $n$  на матрицата  $M$ ;

**fill**( $M; x$ ) - запълва матрицата  $M$  със стойност  $x$ ;  
**fill\_row**( $M; x; i$ ) - запълва  $i$ -тия ред на матрицата  $M$  със стойност  $x$ ;  
**fill\_col**( $M; x; j$ ) - запълва  $j$ -тия стълб на матрицата  $M$  със стойност  $x$ ;  
**copy**( $A; B; i; j$ ) - копира всички елементи от  $A$  в  $B$ , започвайки от индекси  $i$  и  $j$  на  $B$ ;  
**add**( $A; B; i; j$ ) - добавя всички елементи от  $A$  към тези на  $B$ , започвайки от индекси  $i$  и  $j$  на  $B$ ;  
**row**( $M; i$ ) - извлича  $i$ -тия ред на матрицата  $M$  като вектор;  
**col**( $M; j$ ) - извлича  $j$ -тия стълб на матрицата  $M$  като вектор;  
**extract\_rows**( $M; \vec{i}$ ) - извлича онези редове от матрицата  $M$  чиито индекси се съдържат във вектор  $\vec{i}$ ;  
**extract\_cols**( $M; \vec{j}$ ) - извлича онези стълбове от матрицата  $M$  чиито индекси се съдържат във вектор  $\vec{j}$ ;  
**diag2vec**( $\vec{v}$ ) - извлича диагоналните елементи от матрицата  $M$  като вектор;  
**submatrix**( $M; i_1; i_2; j_1; j_2$ ) - извлича подматрица на  $M$ , ограничена от редове  $i_1$  и  $i_2$  и стълбове  $j_1$  и  $j_2$ , вкл.;

#### Данни:

**sort\_cols**( $M; i$ ) - сортира стълбовете на  $M$  на базата на стойностите в ред  $i$  във възходящ ред;  
**rsort\_cols**( $M; i$ ) - сортира стълбовете на  $M$  на базата на стойностите в ред  $i$  в низходящ ред;  
**sort\_rows**( $M; j$ ) - сортира редовете на  $M$  а базата на стойностите в стълб  $j$  във възходящ ред;  
**rsort\_rows**( $M; j$ ) - сортира редовете на  $M$  а базата на стойностите в стълб  $j$  в низходящ ред;  
**order\_cols**( $M; i$ ) - индексите на стълбовете на  $M$ , подредени възходящо по стойностите от ред  $i$ ;  
**revorder\_cols**( $M; i$ ) - индексите на стълбовете на  $M$ , подредени низходящо по стойностите от ред  $i$ ;  
**order\_rows**( $M; j$ ) - индексите на редовете на  $M$ , подредени възходящо по стойностите от стълб  $j$ ;  
**revorder\_rows**( $M; j$ ) - индексите на редовете на  $M$ , подредени низходящо по стойностите от стълб  $j$ ;  
**mcount**( $M; x$ ) - броя на елементите със стойност  $x$  в матрицата  $M$ ;  
**msearch**( $M; x; i; j$ ) - вектор с двата индекса на първия елемент със стойност  $x$  в матрицата  $M$ , започвайки от индекси  $i$  и  $j$ ;  
**mfind**( $M; x$ ) или  
**mfind\_eq**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $= x$ ;  
**mfind\_ne**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $\neq x$ ;  
**mfind\_lt**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $< x$ ;  
**mfind\_le**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $\leq x$ ;

**mfind\_gt**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $> x$ ;  
**mfind\_ge**( $M; x$ ) - индексите на всички елементи в  $M$ , които са  $\geq x$ ;  
**hlookup**( $M; x; i_1; i_2$ ) или  
**hlookup\_eq**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  of  $M$ , за които елементите от ред  $i_1$  са  $= x$ ;  
**hlookup\_ne**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  на  $M$ , за които елементите от ред  $i_1$  са  $\neq x$ ;  
**hlookup\_lt**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  на  $M$ , за които елементите от ред  $i_1$  са  $< x$ ;  
**hlookup\_le**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  на  $M$ , за които елементите от ред  $i_1$  са  $\leq x$ ;  
**hlookup\_gt**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  на  $M$ , за които елементите от ред  $i_1$  са  $> x$ ;  
**hlookup\_ge**( $M; x; i_1; i_2$ ) - стойностите от ред  $i_2$  на  $M$ , за които елементите от ред  $i_1$  са  $\geq x$ ;  
**vlookup**( $M; x; j_1; j_2$ ) или  
**vlookup\_eq**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $= x$ ;  
**vlookup\_ne**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $\neq x$ ;  
**vlookup\_lt**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $< x$ ;  
**vlookup\_le**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $\leq x$ ;  
**vlookup\_gt**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $> x$ ;  
**vlookup\_ge**( $M; x; j_1; j_2$ ) - стойностите от стълб  $j_2$  на  $M$ , за които елементите от стълб  $j_1$  са  $\geq x$ ;

#### Математически:

**hprod**( $A; B$ ) - произведение на Hadamard на матриците  $A$  и  $B$ ;  
**fprod**( $A; B$ ) - произведение на Frobenius на матриците  $A$  и  $B$ ;  
**kprod**( $A; B$ ) - произведение на Kronecker на матриците  $A$  и  $B$ ;  
**mnorm\_1**( $M$ ) - L1 норма на матрицата  $M$ ;  
**mnorm**( $M$ ) или  
**mnorm\_2**( $M$ ) - L2 норма на матрицата  $M$ ;  
**mnorm\_e**( $M$ ) - норма на Frobenius на матрицата  $M$ ;  
**mnorm\_i**( $M$ ) -  $L^\infty$  норма на матрицата  $M$ ;  
**cond\_1**( $M$ ) - число на обусловеност на  $M$  на база на L1 нормата;  
**cond**( $M$ ) или  
**cond\_2**( $M$ ) - число на обусловеност на  $M$  на база на L2 нормата;  
**cond\_e**( $M$ ) - число на обусловеност на  $M$  на база на нормата на Frobenius;

**cond\_i**( $M$ ) - число на обусловеност на  $M$  на база на  $L^\infty$  нормата;  
**det**( $M$ ) - детерминанта на матрицата  $M$ ;  
**rank**( $M$ ) - ранг на матрицата  $M$ ;  
**trace**( $M$ ) - следа на матрицата  $M$ ;  
**transp**( $M$ ) - транспонираната матрица на  $M$ ;  
**adj**( $M$ ) - адюнгираната матрица на  $M$ ;  
**cofactor**( $M$ ) - кофакторната матрица на  $M$ ;  
**eigenvals**( $M$ ) - собствените стойности на матрицата  $M$ ;  
**eigenvecs**( $M$ ) - собствените вектори на матрицата  $M$ ;  
**cholesky**( $M$ ) - декомпозиция на Холецки на симетрична, положително определена матрица  $M$ ;  
**lu**( $M$ ) - LU декомпозиция на матрицата  $M$ ;  
**qr**( $M$ ) - QR декомпозиция на матрицата  $M$ ;  
**svd**( $M$ ) - декомпозиция по особени стойности на  $M$ ;  
**inverse**( $M$ ) - обратната матрица на  $M$ ;  
**lsolve**( $A$ ;  $\vec{b}$ ) - решава системата линейни уравнения  $A\vec{x} = \vec{b}$  чрез LDLT декомпозиция за симетрични матрици и LU декомпозиция за несиметрични;  
**clsolve**( $A$ ;  $\vec{b}$ ) - решава системата линейни уравнения  $A\vec{x} = \vec{b}$  със симетрична, положително определена матрица на коефициентите  $A$  посредством декомпозиция на Холецки;  
**msolve**( $A$ ;  $B$ ) - решава обобщеното матрично уравнение  $AX = B$  чрез LDLT декомпозиция за симетрични матрици и LU декомпозиция за несиметрични;  
**cmsolve**( $A$ ;  $B$ ) - решава обобщеното матрично уравнение  $AX = B$  със симетрична, положително определена матрица на коефициентите  $A$  посредством декомпозиция на Холецки;

#### Двойна интерполация:

**take**( $x$ ;  $y$ ;  $M$ ) - връща елемента на матрицата  $M$  с индекси  $x$  и  $y$ ;  
**line**( $x$ ;  $y$ ;  $M$ ) - двойна линейна интерполация от елементите на  $M$  на база на стойностите на  $x$  и  $y$ ;  
**spline**( $x$ ;  $y$ ;  $M$ ) - двойна spline интерполация на Ермит от елементите на матрицата  $M$  на база на стойностите на  $x$  и  $y$ .

Коментари: "Заглавие" или 'текст', съответно в двойни и единични кавички. Разрешено е използването на **HTML**, **CSS**, **JS** и **SVG** в коментарите.

- Графики на функции:

$\$Plot\{f(x) @ x = a : b\}$  - стандартна, единична;  
 $\$Plot\{x(t) | y(t) @ t = a : b\}$  - параметрична;  
 $\$Plot\{f_1(x) \& f_2(x) \& \dots @ x = a : b\}$  - паралелни;  
 $\$Plot\{x_1(t) | y_1(t) \& x_2(t) | y_2(t) \& \dots @ t = a : b\}$  - паралелни параметрични;  
 $\$Map\{f(x; y) @ x = a : b \& y = c : d\}$  - изохроми на 2D функция;  
 $PlotHeight$  - височина на полето за чертане в пиксели;



*PlotWidth* - ширина на полето за чертане в пиксели;

*PlotStep* - стъпка на мрежата за интерполиране.

- Итеративни и числени методи:

*\$Root*{*f(x) = const @ x = a : b*} - намиране на корен на *f(x) = const*;

*\$Root*{*f(x) @ x = a : b*} - намиране на корен на *f(x) = 0*;

*\$Find*{*f(x) @ x = a : b*} - намира мястото, където функцията пресича абсцисата, но не се изисква стриктно *x* да е решение;

*\$Sup*{*f(x) @ x = a : b*} - локален максимум на функция;

*\$Inf*{*f(x) @ x = a : b*} - локален минимум на функция;

*\$Area*{*f(x) @ x = a : b*} - числено интегриране с адаптивна квадратура на Гаус-Лобато-Кронрод;

*\$Integral*{*f(x) @ x = a : b*} - числено интегриране с Tanh-Sinh квадратура;

*\$Slope*{*f(x) @ x = a : b*} - числено диференциране;

*\$Sum*{*f(x) @ x = a : b*} - крайна сума;

*\$Product*{*f(k) @ k = a : b*} - крайно произведение;

*\$Repeat*{*f(k) @ k = a : b*} - обща итеративна процедура;

*Precision* - точност за числени методи [ $10^{-2}$ ;  $10^{-16}$ ] (по подразбиране -  $10^{-14}$ );

- Условни разклонения:

Стандартно:

*#if* *условие*

*тук въведете код*

*#end if*

Алтернативно:

*#if* *условие*

*тук въведете код*

*#else*

*алтернативен код*

*#end if*

Пълно:

*#if* *условие1*

*тук въведете код*

*#else if* *условие2*

*тук въведете код*

*#else*

*алтернативен код*

*#end if*

Може да добавяте произволен брой "*#else if*" блокове, но само един "*#else*".

- Блок за цикъл:

Стандартен:

*#repeat* *брой повторения*

тук въведете код

**#loop**

Условен:

**#repeat** брой повторения

тук въведете код

**#if** условие

**#break** или **#continue**

**#end if**

още код

**#loop**

- Модули и макроси/текстови променливи:

Модули:

**#include** *име\_на\_файл* - вмъква код от външен файл (модул);

**#local** - начало на локална секция (не се вмъква);

**#global** - начало на глобална секция (вмъква се);

Едноредова текстова променлива:

**#def** *variable\_name*\$ = *съдържание*

Многоредова текстова променлива:

**#def** *variable\_name*\$

*съдържание ред 1*

*съдържание ред 2*

...

**#end def**

Едноредов макрос:

**#def** *macro\_name*\$(*param1*\$; *param2*\$; ...) = *съдържание*

Многоредов макрос:

**#def** *macro\_name*\$(*param1*\$; *param2*\$; ...)

*съдържание ред 1*

*съдържание ред 2*

...

**#end def**

- Контрол на видимостта:

**#hide** - скривай съдържанието на документа;

**#show** - показвай винаги съдържанието (по подразбиране);

**#pre** - показвай следващото съдържание само при въвеждане;

**#post** - показвай следващото съдържание само в резултатите;

**#val** - показвай само изчислените стойности;

**#equ** - показвай пълните формули (по подразбиране);

**#noc** - показвай само формули без стойности (no calculations);

**#nosub** - не замествай стойностите на променливите (no substitution);

**#novar** - показвай само заместените стойности на променливите (no variables);

**#varsub** - показвай формулите с променливи и заместени стойности (по подразбиране);

**#round** *n* - закръглявай до *n* цифри след десетичната точка.

Всяка от горните команди е валидна от мястото на използването и до края на документа или докато не бъде отменена от алтернативна команда.

- Точки на прекъсване (постъпково изпълнение):

**#pause** - изчислява до съответния ред и спира на пауза;

**#input** - показва формуляр за вход на данни до съответния ред и спира на пауза.

- Единици за тригонометрични функции: **#deg** - градуси, **#rad** - радиани, **#gra** - гради;
- Разделител за отправни единици: |;
- Връщай резултати от тригонометр. функции с мерни единици: *ReturnAngleUnits* = 1;
- Бездименсионни единици: %, ‰;
- Единици за ъгли: °, ', ", deg, rad, grad, rev;
- Метрични единици (SI и съвместими):

Маса: g, hg, kg, t, kt, Mt, Gt, dg, cg, mg, µg, ng, pg, Da, u;

Дължина: m, km, dm, cm, mm, µm, nm, pm, AU, ly;

Време: s, ms, µs, ns, ps, min, h, d, w, y;

Честота: Hz, kHz, MHz, GHz, THz, mHz, µHz, nHz, pHz, rpm;

Скорост: kmh;

Електрически поток: A, kA, MA, GA, TA, mA, µA, nA, pA;

Температура: °C, Δ°C, K;

Количество вещество: mol;

Интензитет на светлината: cd;

Площ: a, daa, ha;

Обем: L, daL, hL, dL, cL, mL, µL, nL, pL;

Сила: dyn, N, daN, hN, kN, MN, GN, TN, gf, kgf, tf;

Момент: Nm, kNm;

Налягане: Pa, daPa, hPa, kPa, MPa, GPa, TPa,

dPa, cPa, mPa, µPa, nPa, pPa,

bar, mbar, µbar, atm, at, Torr, mmHg;

Вискозитет: P, cP, St, cSt;

Енергия/работа: J, kJ, MJ, GJ, TJ, mJ, µJ, nJ, pJ,

Wh, kWh, MWh, GWh, TWh, cal, kcal, erg,

eV, keV, MeV, GeV, TeV, PeV, EeV;

Мощност: W, kW, MW, GW, TW, mW, µW, nW, pW, hpM, ks,

VA, kVA, MVA, GVA, TVA, mVA, µVA, nVA, pVA,

VAR, kVAR, MVAR, GVAR, TVAR, mVAR, µVAR, nVAR, pVAR;

Електрически заряд: C, kC, MC, GC, TC, mC, µC, nC, pC, Ah, mAh;

Напрежение: V, kV, MV, GV, TV, mV, µV, nV, pV;

Капацитет: F, kF, MF, GF, TF, mF, µF, nF, pF;

Съпротивление: Ω, kΩ, MΩ, GΩ, TΩ, mΩ, µΩ, nΩ, pΩ;

Проводимост: S, kS, MS, GS, TS, mS,  $\mu$ S, nS, pS,

$\bar{U}$ , k $\bar{U}$ , M $\bar{U}$ , G $\bar{U}$ , T $\bar{U}$ , m $\bar{U}$ ,  $\mu\bar{U}$ , n $\bar{U}$ , p $\bar{U}$ ;

Магнитен поток: Wb, kWb, MWb, GWb, TWb, mWb,  $\mu$ Wb, nWb, pWb;

Плътност на потока: T, kT, MT, GT, TT, mT,  $\mu$ T, nT, pT;

Индукция: H, kH, MH, GH, TH, mH,  $\mu$ H, nH, pH;

Светлинен поток: lm;

Осветеност: lx;

Радиоактивност: Bq, kBq, MBq, GBq, TBq, mBq,  $\mu$ Bq, nBq, pBq, Ci, Rd;

Погълната доза: Gy, kGy, MGy, GGy, TGy, mGy,  $\mu$ Gy, nGy, pGy;

Еквивалентна доза: Sv, kSv, MSv, GSv, TSv, mSv,  $\mu$ Sv, nSv, pSv;

Активност на катализатор: kat;

- Неметрични единици (UK/US):

Маса: gr, dr, oz, lb (или lbm, lb<sub>m</sub>), kipm (или kip<sub>m</sub>), st, qr,

cwt (или cwt<sub>UK</sub>, cwt<sub>US</sub>), ton (или ton<sub>UK</sub>, ton<sub>US</sub>), slug;

Дължина: th, in, ft, yd, ch, fur, mi, ftm (или ftm<sub>UK</sub>, ftm<sub>US</sub>),

cable (или cable<sub>UK</sub>, cable<sub>US</sub>), nmi, li, rod, pole, perch, lea;

Скорост: mph, knot;

Температура: °F,  $\Delta$ °F, °R;

Площ: rood, ac;

Обем, течност: fl\_oz, gi, pt, qt, gal, bbl, or:

fl\_oz<sub>UK</sub>, gi<sub>UK</sub>, pt<sub>UK</sub>, qt<sub>UK</sub>, gal<sub>UK</sub>, bbl<sub>UK</sub>,

fl\_oz<sub>US</sub>, gi<sub>US</sub>, pt<sub>US</sub>, qt<sub>US</sub>, gal<sub>US</sub>, bbl<sub>US</sub>;

Обем, сух: (US) pt<sub>dry</sub>, (US) qt<sub>dry</sub>, (US) gal<sub>dry</sub>, (US) bbl<sub>dry</sub>,

pk (или pk<sub>UK</sub>, pk<sub>US</sub>), bu (или bu<sub>UK</sub>, bu<sub>US</sub>);

Сила: ozf (или oz<sub>f</sub>), lbf (или lb<sub>f</sub>), kip (или kipf, kip<sub>f</sub>), tonf (или ton<sub>f</sub>), pdl;

Налягане: osi, osf, psi, psf, ksi, ksf, tsi, tsf, inHg;

Енергия/работа: BTU, therm (или therm<sub>UK</sub>, therm<sub>US</sub>), quad;

Мощност: hp, hpE, hpS;

- Потребителски единици: .Име = израз.

Имената могат да съдържат и символи за валута: €, £, ₣, ¥, ¢, ₧, ₹, ₩, ₪.

### Готови оразмерителни програми по Еврокод

Разполагаме с богата библиотека от оразмерителни програми за Calcpad по Еврокод, които може да ползвате в готов вид, на символична цена.

Пълен списък от разработените програми, ще намерите на следния линк:

<https://www.proektsoft.bg/calcpad/Pricelist-2024-Calcpad.pdf>

Как да поръчаме?

1. Изберете програмите, които са Ви необходими.
2. Изпратете ни списък с номерата на избраните записки или пакети по имейл.
3. Ще Ви подготвим и изпратим индивидуална оферта.

За заявки, пишете на:

[proektsoft.bg@gmail.com](mailto:proektsoft.bg@gmail.com)

Заплащането е еднократно, без абонамент. Веднъж закупени, програмите могат да се ползват без ограничение.