**Как работи?**

1. **Въведете** текста и формулите в прозореца "**Код**" отляво.
2. Натиснете **F5** или бутона C:\Users\Petar_PC\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B5EC8309.tmp за да **изчислите** резултатите. Те ще се покажат отдясно в прозореца "**Резултати**", като професионално оформена изчислителна **записка**.
3. Натиснете C:\Users\Petar_PC\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\914B22FF.tmp 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** ± **im***i* (например **3** - **2***i*);
* Реални вектори: [*v*1; *v*2; *v*3; ...; *v*n];
* Реални матрици: [*M*11; *M*12; ... ; *M*1*n* | *M*21; *M*22; ... ; *M*2*n* ... | M*m*1; M*m*2; ... ; M*mn*];
* Променливи:  
  - всякакви букви на Unicode;  
  - цифри: **0** - **9**;  
  - запетая: "**,**";  
  - специални символи: **′** , **″** , **‴** , **⁗** , **‾** , **ø** , **Ø** , **°** , **∡**;  
  - горни индекси: **⁰** , **¹** , **²** , **³** , **⁴** , **⁵** , **⁶** , **⁷** , **⁸** , **⁹** , **ⁿ** , **⁺** , **⁻**;  
  - долни индекси: **₀** , **₁** , **₂** , **₃** , **₄** , **₅** , **₆** , **₇** , **₈** , **₉** , **₊** , **₋** , **₌** , **₍** , **₎**;  
  - "**\_**" за долен индекс;  
  Имената на променливи трябва да започват с буква. Различава главни и малки букви.
* Константи:  
  π, e, φ, γ, g, G, ME, MS, c, h, μ0, ε0, ke, e, me, mp, mn, NA, σ, kB, R, F, γc, γs, γa, γg, γw
* 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 (>=);  
  "**∧**" - логическо "И"/AND (&&);  
  "**∨**" - логическо "ИЛИ"/OR (||);  
  "**⊕**" - изключващо "ИЛИ"/XOR (^^);  
  "[**∠**](file:///C:\Users\proek\source\repos\Calcpad\Calcpad.Wpf\doc\help.bg.html#0)" - фазор A∠φ (<<);  
  "**=**" - присвояване;
* Потребителски функции от вида **f**(*x*; *y*; *z*; ... );
* 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;  
   **conj**(*z*)  - спрегнато 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*...;  
   **sumsq**(*x*; *y*; *z*...) - сума от квадратите = *x*² + *y*² + *z*²...;  
   **srss**(*x*; *y*; *z*...) - корен квадратен от сумата на квадратите = **sqrt**(*x*² + *y*² + *z*²...);  
   **average**(*x*; *y*; *z*...) - средно аритметично от множество стойности = (*x* + *y* + *z*...)/n;  
   **product**(*x*; *y*; *z*...) - произведение на множество стойности = *x*·*y*·*z*...;  
   **mean**(*x*; *y*; *z*...) - средно геометрично = n-th **root**(*x*·*y*·*z*...);  
   **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*; *стойност12*;... ; *по-подразб*.) - избирателно 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*;  
   **getunits**(*x*) - връща мерните единици на *x* без числото или 1 ако *x* няма мерни единици;  
   **setunits**(*x*; *u*) - задава мерни единици *u* на *x*, където *x* може да бъде скалар, вектор или матрица;  
   **clrunits**(*x*) - изчиства мерните единици от скалар, вектор или матрица *x*;  
   **hp**(*x*) - преобразува *x* към еквивалентния му високопроизводителен (hp) тип;  
   **ishp**(*x*) - проверява дали типа на *x* е високопроизводителен (hp) вектор или матрица;

Векторни:  
 Създаване и инициализация:  
 **vector**(n) - създава празен вектор с дължина n;  
 **vector\_hp**(n) - създава празен високопроизводителен (hp) вектор с дължина n;  
 **range**(x1; xn; s) - създава вектор от стойностите в интервала от x1 до xn със стъпка s;  
 **range\_hp**(x1; xn; s) - създава високопроизводителен (hp) вектор от стойностите в   
 посочения интервал;  
 Структурни:

**len**( v⃗ ) - връща дължината на вектора v⃗ ;  
 **size**(v⃗ ) - действителния размер на вектора v⃗   
 (индекса на последния ненулев елемент);  
 **resize**(v⃗ ; n) - задава нова дължина n на вектора v⃗ ;  
 **fill**(v⃗ ; x) - запълва вектора v⃗ със стойност x;  
 **join**(A ; b⃗ ; c…) - създава вектор чрез обединяване на аргументите в списъка -   
 матрици, вектори и скалари;  
 **slice**(v⃗ ; i1; i2) -  връща частта от вектора v⃗ , ограничена от индекси i1 и i2, вкл.;  
 **first**( v⃗ ; n) - първите n елемента на вектора v⃗ ;  
 **last**(v⃗ ; n) - последните n елемента на вектора v⃗ ;  
 **extract**(v⃗ ; i ⃗ ) - извлича онези елементи от v⃗ , чиито индекси се съдържат в *i*⃗ ;

Данни:

**sort**(v⃗ ) - сортира вектора v⃗ във възходящ ред;  
 **rsort**(v⃗ ) - сортира вектора v⃗ в низходящ ред;  
 **order**(v⃗ ) - индексите на v⃗ , подредени по възходящ ред на неговите елементи;  
 **revorder**(v⃗ ) - индексите на v⃗ , подредени по низходящ ред на неговите елементи;  
 **reverse**(v⃗ ) - нов вектор, съдържащ елементите на v⃗  в обратен ред;  
 **count**(v⃗ ; x; i) - броя на елементите в v⃗ , от i-тия нататък, които са равни на x ;  
 **search**(v⃗ ; x; i) - индекса на първия елемент в v⃗ , от i-тия нататък, който е равен на x;  
 **find**(v⃗ ; x; i) или  
 **find\_eq**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са = x;  
 **find\_ne**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са ≠ x;  
 **find\_lt**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са < x;  
 **find\_le**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са ≤ x;  
 **find\_gt**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са > x;  
 **find\_ge**(v⃗ ; x; i) - индексите на всички елементи в v⃗ , от i-тия нататък, които са ≥ x;  
 **lookup**(a⃗ ; b⃗ ; x) или  
 **lookup\_eq**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са = x;  
 **lookup\_ne**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са ≠ x;  
 **lookup\_lt**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са < x;  
 **lookup\_le**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са ≤ x;  
 **lookup\_gt**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са > x;  
 **lookup\_ge**(a⃗ ; b⃗ ; x) - всички елементи в a⃗ , за които съответните елементи в b⃗ са ≥ x;

Математически:

**norm\_1**(v⃗ ) - L1 (Манхатън) норма на вектора v⃗ ;  
 **norm**(v⃗ ) или   
 **norm\_2**(v⃗ ) или   
 **norm\_e**(v⃗ ) - L2 (Евклидова) норма на вектора v⃗ ;  
 **norm\_p**(v⃗ ; p) - Lp норма на вектора v⃗ ;  
 **norm\_i**(v⃗ ) - L∞ (безкрайна) норма на вектора v⃗ ;  
 **unit**(v⃗ ) - нормализирания (единичен) вектор v⃗ (с L2 норма = 1);   
 **dot**(a⃗ ; b⃗ ) - скаларно произведение на два вектора a⃗ и b⃗ ;  
 **cross**(a⃗ ; b⃗ ) - векторно произведение на два вектора a⃗ и b⃗ (с дължина 2 или 3);

Матрични:  
 Създаване и инициализация:  
 **matrix**(m; n) - създава празна матрица с размери m⨯n;  
 **identity**(n) - създава единична матрица с размери n⨯n;  
 **diagonal**(n; d) - създава диагонална n⨯n матрица и запълва главния диагонал със стойност d;  
 **column**(m; c) - създава матрица-стълб с размери m⨯1, запълнена със стойност c;  
 **utriang**(n) - създава горна триъгълна матрица с размери n⨯n;  
 **ltriang**(n) - създава долна триъгълна матрица с размери n⨯n;  
 **symmetric**(n) - създава симетрична матрица с размери n⨯n;  
 **matrix\_hp**(m; n) - създава празна високопроизводителна (hp) матрица с размери m⨯n;  
 **identity\_hp** (n) - създава единична високопроизв. (hp) матрица с размери n⨯n;  
 **diagonal\_hp** (n; d) - създава високопроизв. (hp) диагонална n⨯n матрица и   
 запълва диагонала със стойност d;  
 **column\_hp** (m; c) - създава високопроизводителна (hp) матрица-стълб с размери m⨯1, запълнена със стойност c;  
 **utriang\_hp** (n) - създава високопроизв. (hp) горна триъгълна матрица с размери n⨯n;  
 **ltriang\_hp** (n) - създава високопроизв. (hp) долна триъгълна матрица с размери n⨯n;  
 **symmetric\_hp** (n) - създава симетрична матрица с размери n⨯n;

**vec2diag**(v⃗ ) - създава диагонална матрица от елементите на вектора v⃗ ;

**vec2row**(v⃗ ) - създава матрица-ред от елементите на вектора v⃗ ;  
 **vec2col**(v⃗ ) - създава матрица-стълб от елементите на вектора v⃗ ;  
 **join\_cols**( ⃗c1;  ⃗c2;  ⃗c3...) - създава нова матрица чрез обединяване на вектори в стълбове;  
 **join\_rows**( ⃗r1;  ⃗r2;  ⃗r3...) - създава нова матрица чрез обединяване на вектори в редове;  
 **augment**(A; B; C...) - създава нова матрица чрез присъединяване на матриците A, B, C… една до друга;  
 **stack**(A; B; C...) - създава нова матрица чрез присъединяване на матриците A, B, C… 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; i; x) - запълва i-тия ред на матрицата M със стойност x;  
 **fill\_col**(M; j; x) - 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; i⃗ ) - извлича онези редове от матрицата M чиито индекси се  
 съдържат във вектор i⃗ ;  
 **extract\_cols**(M; j⃗ ) - извлича онези стълбове от матрицата M чиито индекси се   
 съдържат във вектор j⃗ ;

**diag2vec**(M) - извлича диагоналните елементи от матрицата M като вектор;  
 **submatrix**(M; i1; i2; j1; j2) - извлича подматрица на M, ограничена от редове i1 и i2 и стълбове j1 и j2, 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 ≠ 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 ≤ 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 ≥ x;  
 **hlookup**(M; x; i1; i2) или

**hlookup\_eq**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са = x;  
 **hlookup\_ne**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са ≠ x;  
 **hlookup\_lt**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са < x;  
 **hlookup\_le**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са ≤ x;  
 **hlookup\_gt**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са > x;  
 **hlookup\_ge**(M; x; i1; i2) - стойностите от ред i2 на M, за които елементите от  
 ред i1 са ≥ x;  
 **vlookup**(M; x; j1; j2) или

**vlookup\_eq**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са = x;  
 **vlookup\_ne**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са ≠ x;  
 **vlookup\_lt**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са < x;  
 **vlookup\_le**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са ≤ x;  
 **vlookup\_gt**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са > x;  
 **vlookup\_ge**(M; x; j1; j2) - стойностите от стълб j2 на M, за които елементите от  
 стълб j1 са ≥ 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\_е**(M) - норма на Frobenius на матрицата M;  
 **mnorm\_i**(M) - L∞ 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\_е**(M) - число на обусловеност на M на база на нормата на Frobenius;  
 **cond\_i**(M) - число на обусловеност на M на база на L∞ 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; b⃗ ) - решава системата линейни уравнения Ax ⃗ = b⃗ чрез LDLT декомпозиция  
 за симетрични матрици и LU декомпозиция за несиметрични;  
 **clsolve**(A; b⃗ ) - решава системата линейни уравнения Ax ⃗ = b⃗ със симетрична,  
 положително определена матрица на коефициентите A посредством   
 декомпозиция на Холецки;  
 **slsolve**(A; b⃗ ) - решава системата линейни уравнения Ax ⃗ = b⃗ с високопроизводителна симетрична, положително определена матрица на коефициентите A чрез метода на преобусловения спрегнат градиент (PCG);  
 **msolve**(A; B ) - решава обобщеното матрично уравнение AX = B чрез LDLT деком-   
 позиция 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;  
 **smsolve**(A; B ) - решава обобщеното матрично уравнение AX = B с високопроизводителна симетрична, положително определена  
 матрица на коефициентите A чрез метода на преобусловения   
 спрегнат градиент (PCG);

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*) & ... @ *x* = *a* : *b* } - паралелни;  
  $Plot { ***x***1(*t*) | ***y***1(*t*) & ***x***2(*t*) | ***y***2(*t*) & ... @ *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;   
*PlotSVG* - чертай графиките във векторен (SVG) формат:  
*PlotAdaptive* - използвай адаптивно сгъстяване на мрежата за графики на функции;  
*PlotStep* - стъпка на мрежата за интерполиране;  
*PlotPalette*  - номер на цветната скала за графика на повърхнина (0-9);  
*PlotShadows*  - чертай повърхнините със светлосенки;   
*PlotSmooth*  - плавно преливане на цветовете (= 1) или изохроми (= 0)   
 за графика на повърхнина;  
*PlotLightDir*  - посока към източника на светлина (0-7) по часовниковата стрелка.

* 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;

$Block { *изрази*  } - многоредов блок от изрази;

$Inline { *изрази* } - едноредов блок от изрази;

[*Precision*](../../../../../../../../../Dev/Calcpad/help.html#0) - 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

С брояч:  
  #for counter = start : end  
    тук въведете код  
  #loop  
С условие:  
  #while условие  
    тук въведете код  
  #loop

* Модули и макроси/текстови променливи:  
  Модули:  
  #include *име\_на\_файл* - вмъква код от външен файл (модул);  
  [#local](file:///C:\Users\Ned\source\repos\Calcpad\Calcpad.Wpf\bin\Release\net6.0-windows\help.html#0) - начало на локална секция (не се вмъква);  
  [#global](file:///C:\Users\Ned\source\repos\Calcpad\Calcpad.Wpf\bin\Release\net6.0-windows\help.html#0) - 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

Текстови/CSV файлове:  
#read *M* from filename.txt@R1C1:R2C2 TYPE=R SEP=',' - четена на матрица *M* от текстов/CSV файл;  
#write *M* to filename.txt@R1C1:R2C2 TYPE=N SEP=',' - запис на матрица *M* в текстов/CSV файл;  
#append *M* to filename.txt@R1C1:R2C2 TYPE=N SEP=',' - добавяне на матрица *M* към текстов/CSV файл;  
Excel файлове (xlsx и xlsm):  
#read *M* from filename.xlsx@Sheet1!A1:B2 TYPE=R - четене на матрица *M* от Excel файл;  
#write *M* to filename.xlsx@Sheet1!A1:B2 TYPE=N - запис на матрица *M* в Excel файл;  
#append *M* to filename.xlsx@Sheet1!A1:B2 TYPE=N - добавяне на матрица *M* към Excel файл;  
Sheet, range, TYPE и SEP могат да бъдат пропуснати.  
За командата #read, TYPE може да бъде някое от [R|D|C|S|U|L|V].  
За командите #write и #append, TYPE може да бъде Y или N.

* 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);  
  #split  - разделяй уравнения, които не се събират на един ред;  
  #wrap  - свивай уравнения които не се събират на един ред (по подразбиране);  
  #round *n*  - закръглява изходните стойности до *n* цифри след десетичната точка;  
  #round default - възстановява закръгляването по подразбиране;  
  #format *FFFF* - задава потребителски форматиращ низ;  
  #format default - възстановява форматирането по подразбиране;  
  #md on  - включва използването на markdown в коментари;  
  #md off  - изключва използването на markdown в коментари;  
  #phasor  - задава изходното форматиране на комплексни числа като полярен фазор: A∠φ;  
  #complex  - задава изходното форматиране на комплексни числа в алгебричен формат: a + bi.  
  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](file:///C:\Users\Ned\source\repos\Calcpad-bg\Calcpad.Wpf\bin\Release\net7.0-windows\help.html#0) - изчислява до съответния ред и спира на пауза;  
  [#input](file:///C:\Users\Ned\source\repos\Calcpad-bg\Calcpad.Wpf\bin\Release\net7.0-windows\help.html#0) - 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: %, ‰, ‱, pcm, ppm, ppb, ppt, ppq;
* 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, μS, nS, pS,   
   ℧, k℧, M℧, G℧, T℧, m℧, μ℧, n℧, p℧;  
  Магнитен поток: Wb , kWb, MWb, GWb, TWb, mWb, μWb, nWb, pWb;  
  Плътност на потока: T, kT, MT, GT, TT, mT, μT, nT, pT;  
  Индукция: H, kH, MH, GH, TH, mH, μH, nH, pH;  
  Светлинен поток: lm;  
  Осветеност: lx;  
  Радиоактивност: Bq, kBq, MBq, GBq, TBq, mBq, μBq, nBq, pBq, Ci, Rd;  
  Погълната доза: Gy, kGy, MGy, GGy, TGy, mGy, μGy, nGy, pGy;  
  Еквивалентна доза: Sv, kSv, MSv, GSv, TSv, mSv, μSv, nSv, pSv;  
  Активност на катализатор: kat;
* Неметрични единици (UK/US):  
  Маса: gr, dr, oz, lb (или lbm,  lb**\_m**), kipm (или kip**\_m**), st, qr,   
   cwt (или  cwt**\_UK**, cwt**\_US**), ton (или ton**\_UK**, ton**\_US**), slug;  
  Дължина: th, in, ft, yd, ch, fur, mi, ftm (или ftm**\_UK**, ftm**\_US**),   
   cable (или cable**\_UK**, cable**\_US**), nmi, li, rod, pole, perch, lea;  
  Скорост: mph , knot;  
  Температура: °F, Δ°F, °R;  
  Площ: rood, ac;  
  Обем, течност: fl\_oz, gi, pt, qt, gal, bbl, or:  
   fl\_oz**\_UK**, gi**\_UK**, pt**\_UK**, qt**\_UK**, gal**\_UK**, bbl**\_UK**,  
   fl\_oz**\_US**, gi**\_US**, pt**\_US**, qt**\_US**, gal**\_US**, bbl**\_US**;   
  Обем, сух: (US) pt**\_dry**, (US) qt**\_dry**, (US) gal**\_dry**, (US) bbl**\_dry**, pk (или pk**\_UK**, pk**\_US**), bu (или bu**\_UK**, bu**\_US**);  
  Сила: ozf (или oz**\_f**), lbf (или lb**\_f**), kip (или kipf, kip**\_f**), tonf (или ton**\_f**), pdl;  
  Налягане: osi, osf, psi, psf, ksi, ksf, tsi, tsf, inHg;  
  Енергия/работа: BTU, therm (или  therm**\_UK**, therm**\_US**), 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 символични цени.

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

<https://www.proektsoft.bg/calcpad/Pricelist-2025-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.