Code utility

Code

Sintaxis

```
test(a,b#) // fn name
// Code
x:a+1
y:b#-1 // subs # by local
cond:1

// while loop
while cond1
  // for loop
  for k,rng
    r:19
    s:20
  // if clause
  if cond2
    t:21
    u:22
  else
    v#:23

m:mat( ... // continue
 x,y, ...  // in
 r,s, ...  // next line
 2,2)

m+v#
```

$test := 1$

Sintaxis Code take the script in the description and try to convert it into a SMath function

$C := Code(\text{description}(test))$

$C = \text{"line(test(a,b\_local\_test):line(x:a+1,y:b\_local\_test-1,cond:1,while(c}$

$test(a, b\_local\\_test) :=$
$x := a + 1$
$y := b\_local\\_test - 1$
$cond := 1$
while $cond1$
  for $k \in rng$
    $r := 19$
    $s := 20$
  if $cond2$
    $t := 21$
    $u := 22$
  else
    $v\_local\\_test := 23$
$m := \begin{bmatrix} x & y \\ r & s \end{bmatrix}$
$m + v\_local\\_test$

Examples

```
RREF(M) // Rew reduced echelon form
stack( A:M, B:A )
stack( m:rows(A), n:cols(A) )
stack( r:1, c:1)
ro:range(1,m)
while (r≤m)&(c≤n)
  if el(A,r,c)≡0
    h:range(r,m)
    el(A,h,c):0
    c:c+1
  else
    k:range(c,n)
    f:el(A,r,c)
    el(A,r,k):el(A,r,k)/f
    el(B,ro,k):el(A,ro,k)- ...
      if(ro≠r,el(A,ro,c)*el(A,r,k),0)
    el(A,ro,k):el(B,ro,k)
    stack( r:r+1, c:c+1)
A
```

$RREF := 1$

$\text{str2num}(Code(\text{description}(RREF))) = 1$

$M := \text{augment}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \text{identity}(3)\right)$

$RREF(M) = \begin{bmatrix} 1 & 0 & -1 & 0 & -2.6667 & 1.6667 \\ 0 & 1 & 2 & 0 & 2.3333 & -1.3333 \\ 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$

```
Bisection(f(1),α,β,ε)
// Bisection method

stack( a:α, b:β )
stack( fa:f(a), fb:f(b) )

if sign(fa)*sign(fb)>0
  error("Wrong interval")
else
  if fa≡0
```

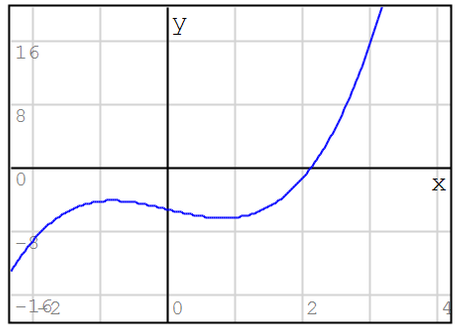
$\text{str2num}(Code(\text{description}(Bisection))) = 1$

$f(x) := x^3 - 2 \cdot x - 5$

$Bisection(f(x), 1, 4, 10^{-7}) = 2.0946$

```
    c:a
  else
    if fb≡0
      c:b
    else
      N:1+trunc(log(abs({b-a}/ε),2))
      for iter, range(1,N)
        c:{a+b}/2
        fc:f(c)
        if fc≡0
          break
        else
          if sign(fa)*sign(fc)>0
            a:c
            fa:fc
          else
            b:c
            fb:fc
c
```

Bisection := 1



$f(x)$

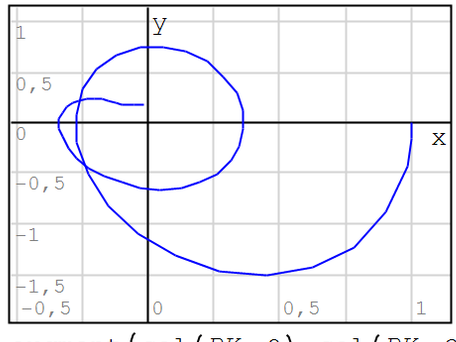
```
RK(D#(2),xo#,to#,te#,N#,ε#)

// Runge-Kutta method
// with adaptive step

X#:xo#
h#:{te#-to#}/{N#+1}
for t#:to#+h#,t#≤te#+to#+h#,t#:t#+h#
  x#:col(X#,cols(X#))
  k1#:h#/3*eval(D#(t#,x#))
  k2#:h#/3*eval(D#(t#+h#/3,x#+k1#))
  k3#:h#/3*eval(D#(t#+h#/3,x#+{k1#+k2#}/2))
  k4#:h#/3*eval(D#(t#+h#/2,x#+{3*k1#+9*k3#}/8))
  k5#:h#/3*eval(D#(t#+h#,x#+{3*k1#-9*k3#}/2+6*k4#))
  δ#:k1#-9/2*k3#+4*k4#-1/2*k5#
  if normi(δ#)≤5*ε#
    X#:augment(X#,x#+1/2*(k1#+4*k4#+k5#))
    el(T#,1,cols(X#)):t#
    h#:h#*2
  else
    t#:t#-h#
    h#:h#/2
transpose(stack(T#,X#))
```

RK := 1

$\text{str2num}\left(\textit{Code}\left(\text{description}\left(\textit{RK}\right)\right)\right) = 1$

$$D(t, x) := \begin{bmatrix} x_2 \\ -\left(5 \cdot x_1 + x_2 - \sin(t)\right) \end{bmatrix}$$

$$RK := RK\left(D(t, x), \begin{bmatrix} 1 \\ 0 \end{bmatrix}, 0, 6, 200, 10^{-5}\right)$$



$\text{augment}\left(\text{col}\left(RK, 2\right), \text{col}\left(RK, 3\right)\right)$

Alvaro