```
PDEParabImplPC := proc (n, m, k, a, b, g, e, f, alfa1, beta1, alfa2, beta2, gama1, gama2, phi)
    local h, n1, m1, u, i, j, alfa, beta, x, F, d1, d2, d3, p, q, r, s, pom, up, df, v;
    with(linalg);
    h := (b-a)/n;
    n1 := n+1;
    m1 := m+1;
    alfa := evalf(k/(h*h));
    beta := evalf(k/(2*h));
    df := D[3](f);
    x := Vector(n1, i -> a+(i-1)*h);
    u := Matrix((m1, n1), 0);
    d1 := Vector(n1, 0);
    d2 := Vector(n1, 0);
    d3 := Vector(n1, 0);
    F := Vector(n1, 0);
    v := Vector(n1, 0);
    #výpočet profilu u - 0 vrstva
    for i from 1 to n1 do
        u[1, i] := evalf(phi(x[i]));
    end do;
    #Výpočet diagonál v matici A a pravé strany, u(j+1)=F(j)+A*u(j)
    for j from 1 to m do
        #výpočet pomocného profilu v pomocí expl. metody - predictor
        for i from 1 to n1 do
            d1[i] := evalf(g(x[i], (j-1)*k)*alfa-e(x[i], (j-1)*k)*beta);
            d2[i] := evalf(1-2*alfa*g(x[i], (j-1)*k));
            d3[i] := evalf(g(x[i], (j-1)*k)*alfa+e(x[i], (j-1)*k)*beta);
            F[i] := evalf(k*f(x[i], (j-1)*k, u[j, i]))
        end do;
        for i from 2 to n do
            v[i] := evalf(F[i]+d1[i]*u[j, i-1]+d3[i]*u[j, i+1]+d2[i]*u[j, i])
        end do;
        # výpočet j+1-ního profilu pomocí implicitní metody - corrector
        # napočítání diagonál matice pravé strany
        for i from 1 to n-1 do
            d1[i] := -evalf(g(x[i+1], j*k)*alfa-e(x[i+1], j*k)*beta);
            d2[i] := evalf(1+2*alfa*g(x[i+1], j*k)-k*df(x[i+1], j*k, u[j, i+1]));
            d3[i] := -evalf(g(x[i+1], j*k)*alfa+e(x[i+1], j*k)*beta);
            F[i] := evalf(u(j, i+1)+k*f(x[i+1], j*k, v[i+1]));
        end do;
        # Dosezení okrajových podmínek do první a poslední rovnice
        p := 1/(alfa1*2*h-3*beta1(j*k));
        r := beta1(j*k)*p;
        pom := d1[1]*r;
        d2[1] := d2[1]-4*pom;
        d3[1] := d3[1]+pom;
        F[1] := F[1]-2*h*gama1(j*k)*d1[1]*p;
        q := 1/(alfa2*2*h+3*beta2(j*k));
        s := beta2(j*k)*q;
        pom := d3[n-1]*s;
        d2[n-1] := d2[n-1]+4*pom;
        d1[n-1] := d1[n-1]-pom;
```

```
        F[n-1] := F[n-1]-2*h*gama2(j*k)*d3[n-1]*q;
        up := TriDiagonalSolve(n-1, d1, d2, d3, F);
        for i from 2 to n do
            u[j+1, i] := up[i-1];
        end do;
        # výpo¦et okrajových podmínek - j+1 vrstva, tj. pro t=j*k
        u[j+1, 1] := 2*h*gama1(j*k)*p-r*(4*u[j+1, 2]-u[j+1, 3]);
        u[j+1, n1] := 2*h*gama2(j*k)*q+s*(4*u[j+1, n]-u[j+1, n-1]);
    end do;
    RETURN(eval(u));
end proc:
```

PŚíklad 1:

```
phi := x→sin(evalf(Pi·x)):
alfa1 := 1:
 beta1 := t→0:
alfa2 := 1:
 beta2 := t→0:
gama1 := t→0:
 gama2 := t→0:
g := (x, t)→1:
 e := (x, t)→0:
f := (x, t, y)→0:
 n := 10:
m := 80:
 k := 0.005:
h := 1.0/n;
 T := k·m;
```

$$0.1000000000$$

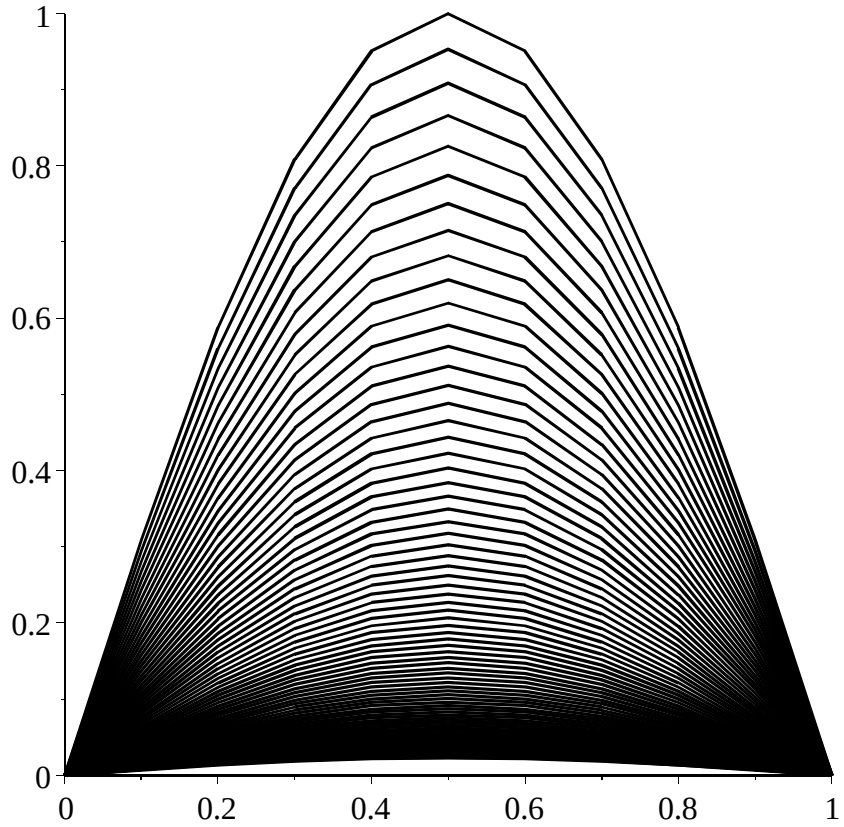
$$0.400 \qquad \textbf{(1.1)}$$

```
vys := PDEParabImplPC(n, m, k, 0.0, 1.0, g, e, f, alfa1, beta1, alfa2, beta2, gama1, gama2, phi);
```

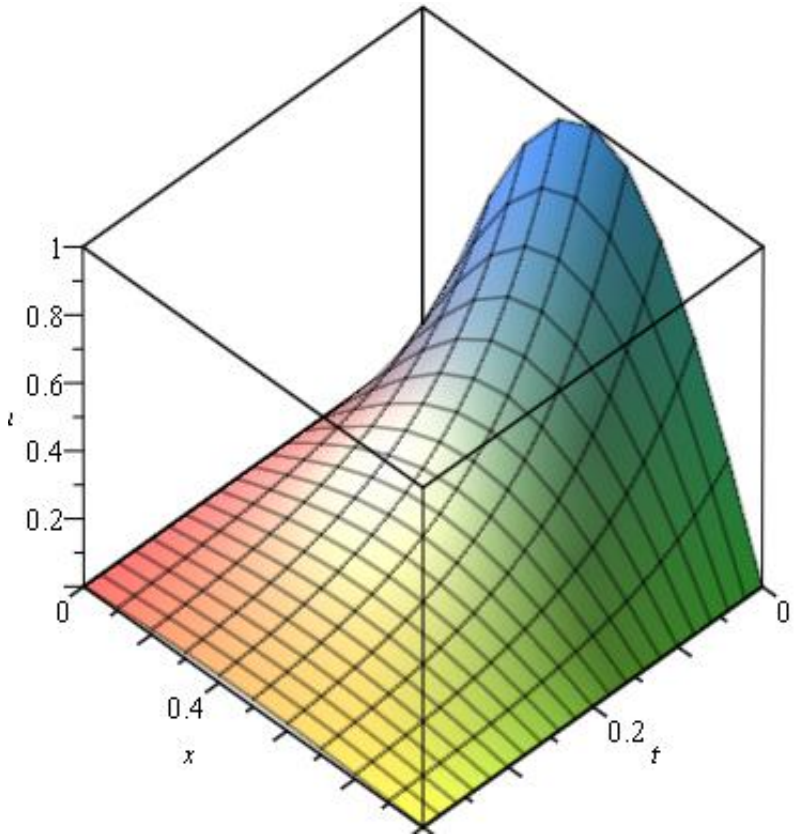
$$\begin{bmatrix} 81 \text{ x } 11 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \qquad \textbf{(1.2)}$$

```
data := [seq([seq([0+(i-1)·h, vys[j, i]], i = 1..n+1)], j = 1..m+1)]:
with(plots):
```

```
display(seq(listplot(data[i]), i = 1..m));
```




```
a := subs(1 .. m + 1 = 0 ..T, 1 .. n + 1 = 0 .. 1, matrixplot(vys[1 .. m + 1, 1 .. n + 1], labels = [t,
     x, u])) :
display(a, view = [0 .. T, 0 .. 1, 0 .. 1]);
```



PŚíklad 2:

phi := $x \to 1 - x \cdot x$:
alfa1 := 1 :
beta1 := $t \to 0$:
alfa2 := 1 :
beta2 := $t \to 0$:
gama1 := $t \to 1$:
gama2 := $t \to 0$:
$g := (x, t) \to 1$:
$e := (x, t) \to 1$:
$f := (x, t, y) \to -\exp(-y)$:
$n := 10$:
$m := 50$:
$k := 0.005$:
$h := \frac{1.0}{n}$;
$T := k \cdot m$;

$$0.1000000000$$

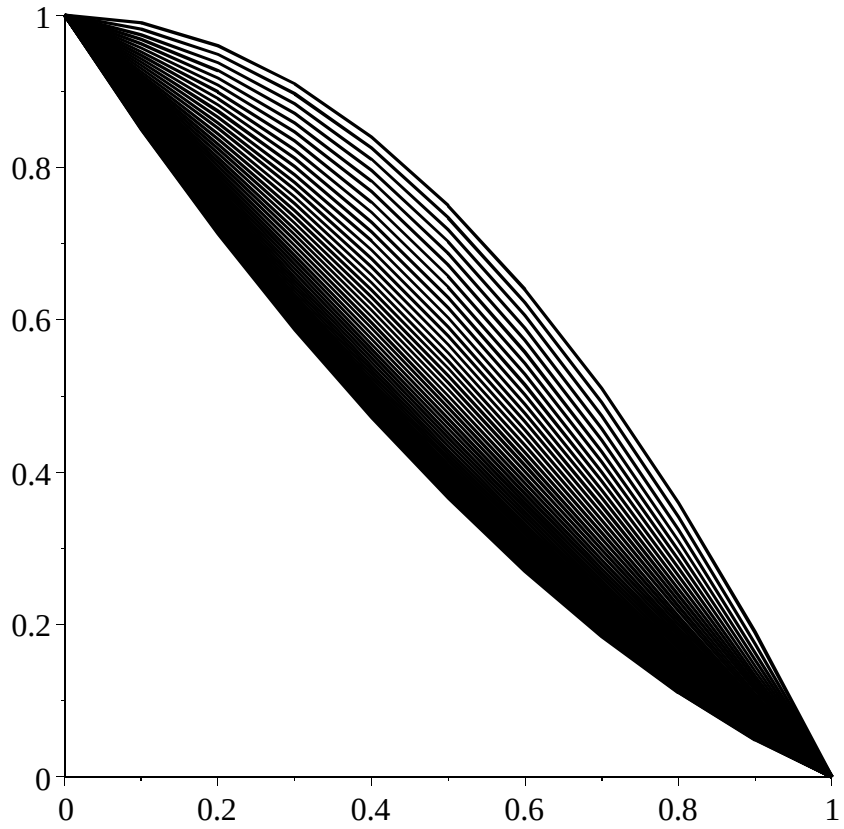
$$0.250 \qquad \textbf{(2.1)}$$

$vys := PDEParabImplPC(n, m, k, 0.0, 1.0, g, e, f, alfa1, beta1, alfa2, beta2, gama1, gama2, \mathrm{phi});$

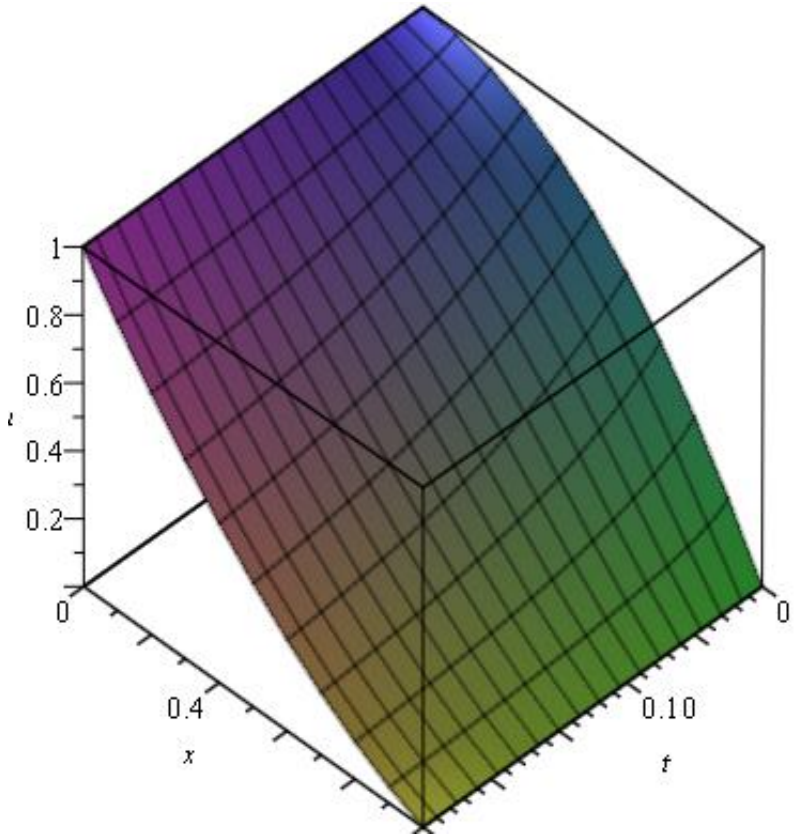
$$\begin{bmatrix} \textit{51 x 11 Matrix} \\ \textit{Data Type: anything} \\ \textit{Storage: rectangular} \\ \textit{Order: Fortran\_order} \end{bmatrix} \qquad \textbf{(2.2)}$$

$data := [seq([seq([0 + (i - 1) \cdot h, vys[j, i]], i = 1 .. n + 1)], j = 1 .. m + 1)]:$
$with(plots):$

$display(seq(listplot(data[\mathrm{i}]), i = 1 .. m));$



$a := subs(1 .. \mathrm{m} + 1 = 0 .. \mathrm{T}, 1 .. \mathrm{n} + 1 = 0 .. 1, matrixplot(vys[1 .. \mathrm{m} + 1, 1 .. \mathrm{n} + 1], labels = [t, x, u]))):$
$display(a, view = [0 .. \mathrm{T}, 0 .. 1, 0 .. 1]);$


1
0.8
0.6
u
0.4
0.2
0
0.4
x
0.10
t
0