

# Informatyka i komputerowe wspomaganie prac inżynierskich

Dr Zbigniew Kozioł - wykład  
Dr Grzegorz Górski - laboratorium

## Wykład III

**Numeryczne rozwiązywanie równań różniczkowych.  
MES, Metoda Elementów Skończonych  
(FEM, Finite Element Method).**

## Dlaczego równania różniczkowo-całkowe oraz FEM?:

- są to bowiem najszerszej spotykane i stosowane kategorie problemów w inżynierskim modelowaniu komputerowym

Przykłady: rozchodzenie się ciepła w turbinie silnika odrzutowego, wnikanie tlenu do stali i zjawisko korozji, wnikanie domieszek do półprzewodnika – **równania dyfuzji**.

Rozkład pola elektromagnetycznego wokół anteny nadawczej, rozkład pola elektrycznego w naczyniu z elektrolitem – **równania Maxwella**

Drgania membrany piezoelektrycznej – **równania Laplaca** opisujące naprężenia w materiale

Większość obliczeń sprowadza się do całkowania i różniczkowania

Wygodną metodą do tego jest metoda elementów skończonych (MES, FEM)

## Przykład numerycznego całkowania (w Python):

[integral00.py](#) ← link

## Przykład numerycznego różniczkowania:

[differential00.py](#) ← link

## Co jest wspólne w obu przykładach?

Tworzenie sieci (mesh) punktów w przestrzeni współrzędnych, na których wykonywane są obliczenia.

Wykorzystajmy tę metodę do rozwiązania skomplikowanego równania różniczkowego,

nieliniowego równania dyfuzji

## Opis problemu: nieliniowe równanie dyfuzji:

$$\frac{\partial \beta}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1}$$

### Warunki:

Przyjmujemy, idealizując problem, iż materiał rozciąga się od minus do plus nieskończoności w kierunkach y-z oraz od -1 do +1 w kierunku x.

Przewodnictwo materiału jest nieliniową, potęgową funkcją prądu. Wtedy wnikanie pola magnetycznego do materiału opisywane jest powyższym cząstkowym równaniem różniczkowym.

**Warunki brzegowe:** { to istotna część definicji problemu przy rozwiązywaniu równań różniczkowych }

W chwili  $t=0$  pole magnetyczne wewnątrz materiału = 0;

W chwili  $t=0$  przykładany nagle pole magnetyczne, równoległe do powierzchni, o wartości  $B=1$ .

## Sposób na numeryczne rozwiązanie nieliniowego równania dyfuzji. Jak sformułować problem numerycznie?

$$\frac{\partial \beta}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1}$$

Zapiszmy powyższe równanie w formie częściowo różnicowej (**NIE** różniczkowej):

Albo w formie tylko **różnicowej**:

$$\Delta \beta = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1} \cdot \Delta t$$

$$\Delta \beta = \frac{\Delta}{\Delta x} \left( \frac{\Delta \beta}{\Delta x} \right)^{\kappa+1} \cdot \Delta t$$

Stwórzmy siatkę punktów:

Wzdłuż współrzędnej  $x$  będą indeksowane wskaźnikami  $i$

Wzdłuż współrzędnej  $t$  będą indeksowane wskaźnikami  $j$

Wtedy możemy zapisać ostatnie równanie w ten sposób:

$$\beta[i, j + 1] = \beta[i, j] + a \cdot \left( (\beta[i + 1, j] - \beta[i, j])^{\kappa+1} - (\beta[i, j] - \beta[i - 1, j])^{\kappa+1} \right)$$

## Nieliniowe równanie dyfuzji: kod w Pascalu

```
FOR ts:=0 TO maxTsteps DO { dwie pętle FOR...TO, jedna wewnątrz drugiej }
BEGIN { f[xs,ts] to tablica przechowująca chwilowe wartości pola magnetycznego, o rozmiarze maxXsteps x maxTsteps }
  FOR xs:=0 TO (maxXsteps) DO
  BEGIN
    f[xs,ts]:=0; { xs zmienia się od 0 do maxXsteps, a tx od 0 do maxTsteps }
  END;
END;

FOR ts:=0 TO maxTsteps DO BEGIN f[0,ts]:=1; f[maxXsteps,ts]:=1; END; { Na brzegach ustawiamy wartość równą 1 }

step:=0;
REPEAT { główna pętla obliczeniowa }
  FOR ts:=0 to maxTsteps-1 DO
  BEGIN
    FOR xs:=1 to (maxXsteps-1) DO
    BEGIN { obliczamy pochodną }
      f[xs,ts+1]:=f[xs,ts]+a*( pwr(f[xs+1,ts] - f[xs,ts], kappa+1) -pwr(f[xs,ts] - f[xs-1,ts], kappa+1) );
    END;
  END;

  FOR xs:=0 TO maxXsteps DO f[xs,0]:=f[xs,maxTsteps]; { na wszelki wypadek; chyba niepotrzebne }

  IF (step=19) OR (step=49) THEN
    SaveBeta((step+1)*maxTsteps); { zapisujemy dane na dysku }

  inc(step);
UNTIL step=maxSTEP; { koniec głównej pętli obliczeniowej }
```

Istnieje pracująca wersja tego kodu w języku Python:

[dyfuzja.py](#) ← link

A także wersja w JavaScript:

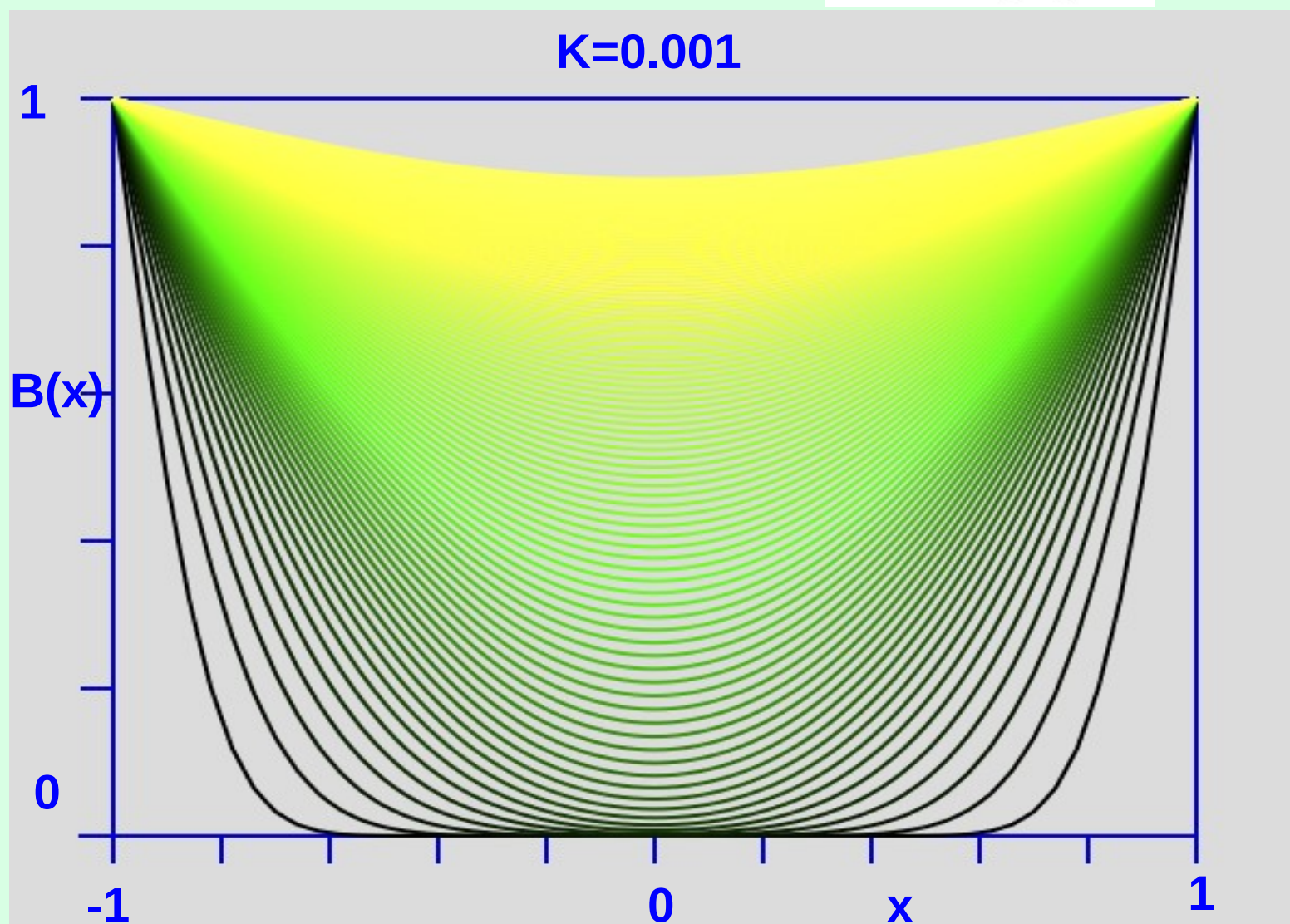
[dyfuzja.html](#) ← link

By samemu popробować zabawy z kodem w JavaScript, trzeba zapisać kopię kodu `dyfuzja.html` na dysku własnego komputera, a następnie zmieniać parametry w kodzie i odświeżać zawartość strony



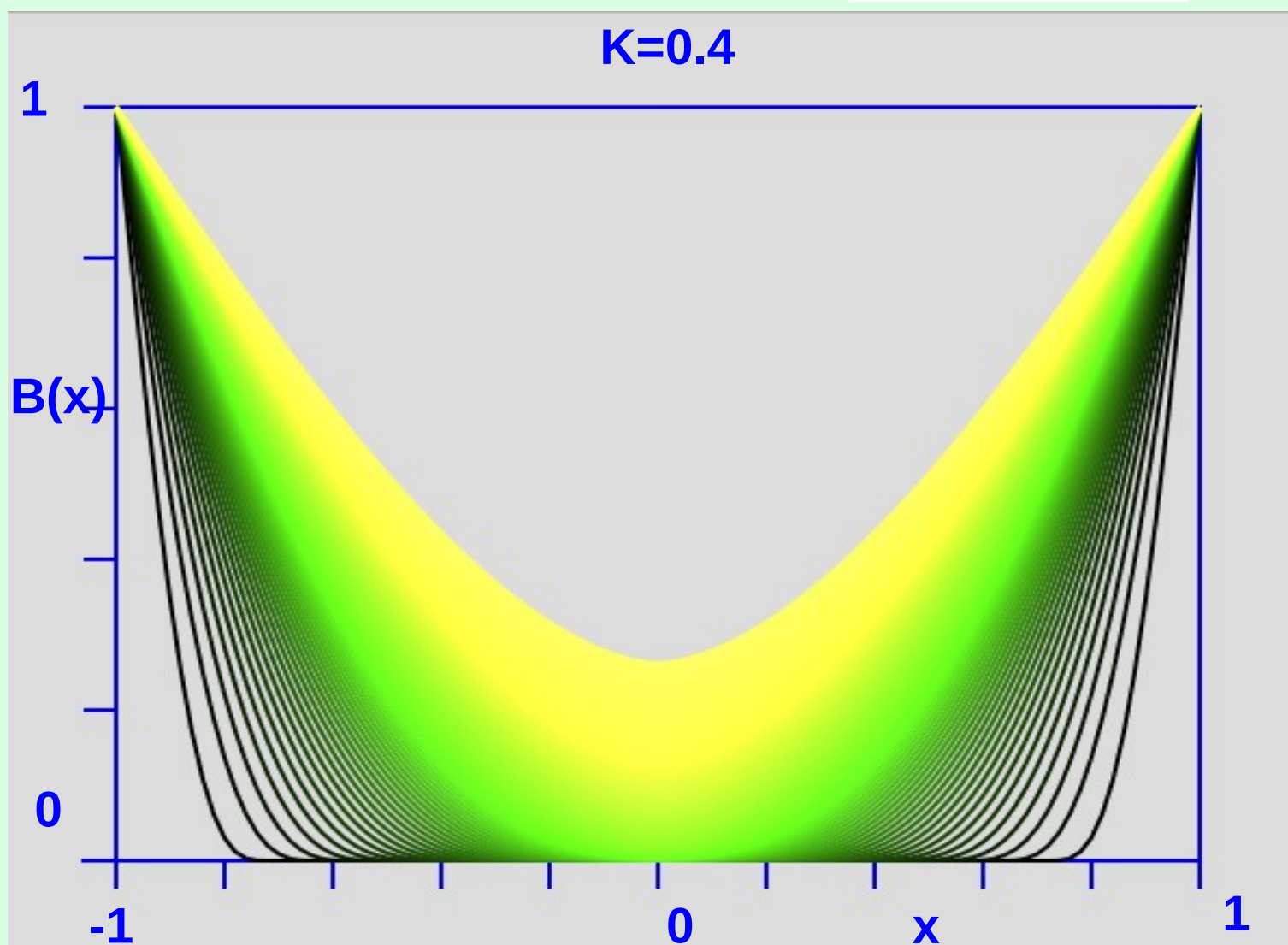
## Rozwiązania równania dyfuzji

$$\frac{\partial \beta}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1}$$



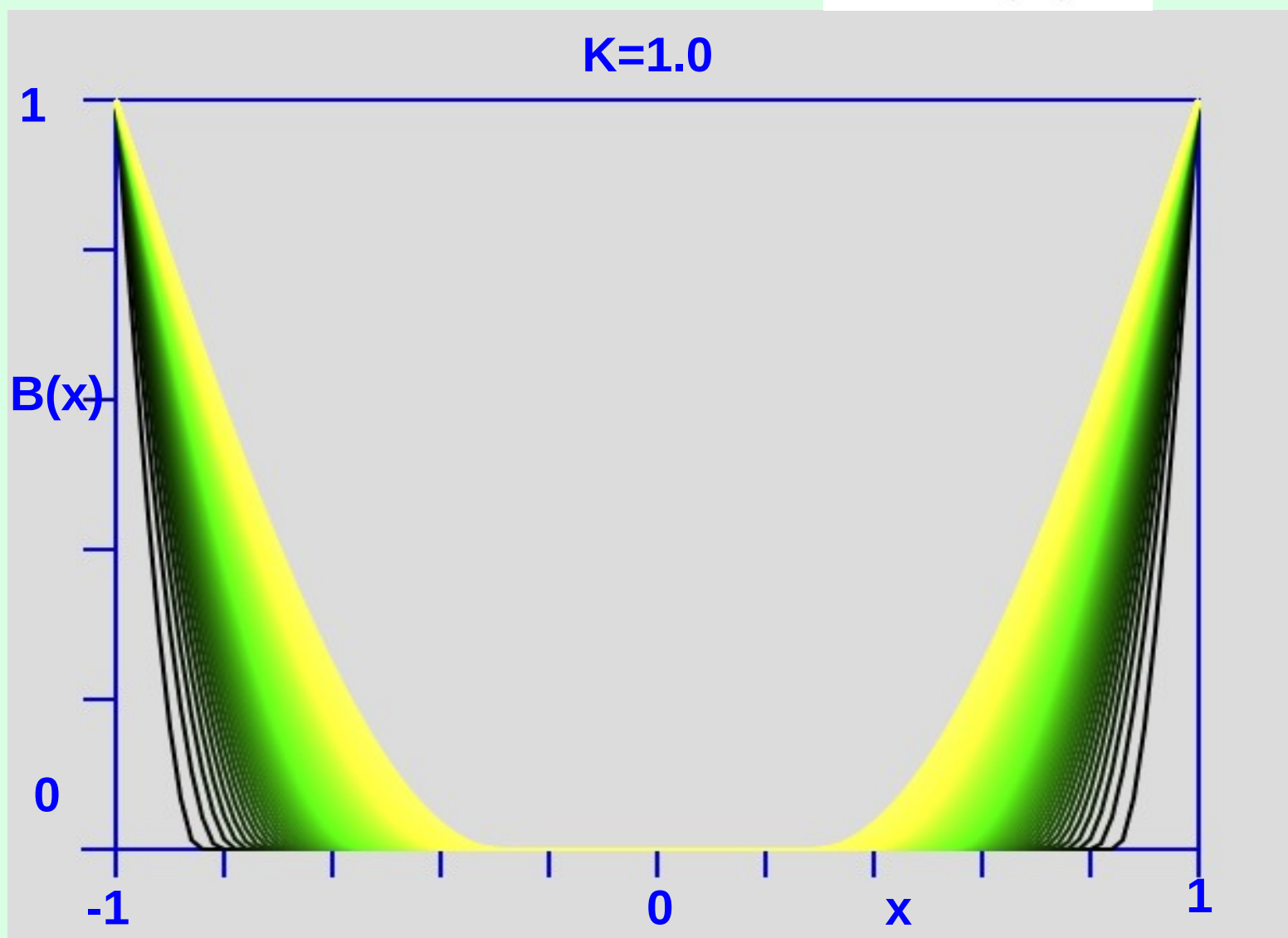
## Rozwiązania równania dyfuzji

$$\frac{\partial \beta}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1}$$



## Rozwiązania równania dyfuzji

$$\frac{\partial \beta}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial \beta}{\partial x} \right)^{\kappa+1}$$



## W jaki sposób zwiększyć dokładność obliczeń?

1. Więcej punktów sieci. Ale to prowadzi szybko do spowolnienia obliczeń.
2. Do obliczania pochodnych czy całek przybliżamy wartości funkcji w sąsiednich punktach przy pomocy dopasowania prostych funkcji, choćby funkcji parabolicznej. Dla prostych funkcji znamy ich pochodne lub całki. Przykład:

Pierwsze przybliżenie:

```
BEGIN { obliczamy pochodną }  
    f[xs,ts+1]:=f[xs,ts]+a*( pwr(f[xs+1,ts] - f[xs,ts], kappa+1) -pwr(f[xs,ts] - f[xs-1,ts], kappa+1) );  
END;
```

Lepsze przybliżenie z wykorzystaniem funkcji parabolicznej(następna strona):

## W jaki sposób zwiększyć dokładność obliczeń?

Lepsze przybliżenie z wykorzystaniem funkcji parabolicznej:

Mamy trzy punkty ( $x_1, x_2, x_3$ ) odległe o  $Dx$ . Wartości funkcji w tych punktach są  $f_1, f_2, f_3$ . Znajdujemy równanie paraboli przechodzącej przez te punkty:  $f(x) = \text{alf} * x^2 + \text{bet} * x + c$

PASCAL

```
FUNCTION alpha(f1, f2, f3, DeltaX: Double): Double;
```

```
BEGIN
```

```
    alpha := (f3-2*f2 + f1)/(2*DeltaX*DeltaX);
```

```
END;
```

```
FUNCTION beta(f2, f3, x2, al, DeltaX: Double): Double;
```

```
BEGIN
```

```
    beta := (f3-f2)/DeltaX - al * (x2*2 + DeltaX);
```

```
END;
```

Pochodna  $f(x)$  będzie  $2*\text{alf}*x + \text{bet}$  . Dokładniejszy kod:

```
al:=alpha(f[xs-1,ts], f[xs,ts], f[xs+1,ts], Dx);
```

```
bet:=beta(f[xs,ts], f[xs+1,ts], xs*Dx, al, Dx);
```

```
deriv:= 2*al*xs*Dx+bet;
```

```
f[xs,ts+1]:=f[xs,ts]+a*( pwr(deriv, kappa))*al;
```

## W jaki sposób zwiększyć jeszcze bardziej dokładność obliczeń?

Zamiast trzech punktów sąsiednich stosować pięć, siedem, lub więcej, dla przybliżania funkcji. Stosować można wielomiany Legendre'a, itp.

W przypadku większych pochodnych konieczne jest stosowanie większej ilości punktów.

## Jednakowoż ...

Przykład tutaj przedstawiony, pewnej metody rozwiązywania nieliniowego równania dyfuzji, nadaje się do rozwiązywania tylko pewnej szczególnej klasy problemów.

W przypadku ogólniejszym, równania różniczkowe czy całkowe rozwiązuje się numerycznie w metodzie MSE (FEM) całkiem inaczej, poprzez rozwiązywanie równań algebraicznych, operacje na macierzach.

O tym w następnym odcinku...