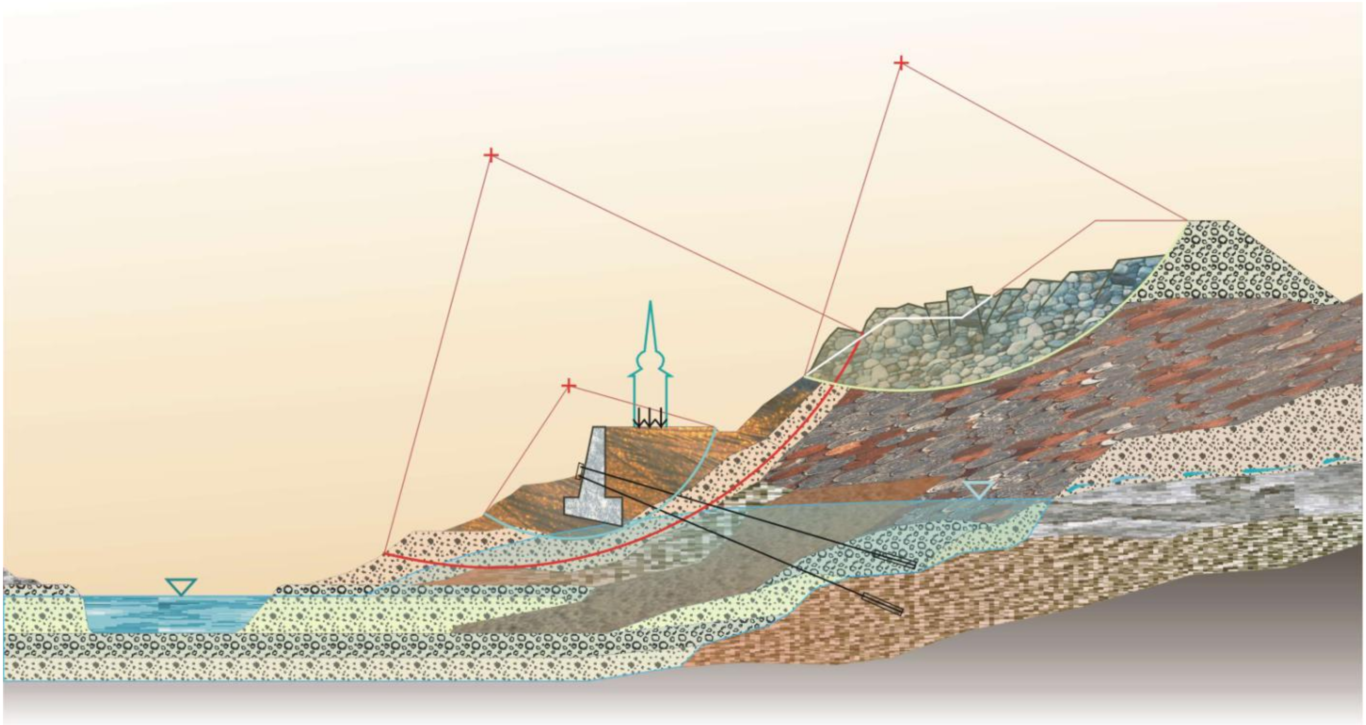
ALGORITMIZÁCIA STABILITY SVAHU

Vladimír Budinský *



„Katastrofálne zosuvy na švédskych železniciach dali podnet k založeniu zvláštnej geotechnickej komisie vo Švédsku v roku 1914. Štúdie tejto komisie boli základom nového vedného odboru mechaniky zemín“. [1]

Prvé exaktné riešenie pochádza z roku 1916, keď Sven Hultin a Knut Petterson vyšetrovali šmykovú plochu prístavnej hrádze v Göteborgu. Verejnosti sa dostalo do povedomia pod označením **průžková metóda s valcovou šmykovou plochou**. Roku 1926 Wolmar Fellenius rozšíril a teoreticky odvodil pravidlá tejto metódy, ktorá je ešte aj dnes uprednostňovaná pre svoju jednoduchosť a univerzálnosť.

Konečnú podobu vzorcom tak, ako sa používajú dnes, dal až Karl Terzaghi roku 1936. U nás sa ujala pod názvom **Pettersonova průžková metóda**. Táto metóda je založená na rovnováhe elementárneho průžku, na ktoré je rozdelená zosúvajúca sa hmota. Ide všeobecne o rovinnú úlohu. Pettersonova metóda zanedbáva silové pôsobenie susedných průžkov navzájom, čo pri viac zakrivených plochách vedie k určitej chybe. Preto sa viacerí ďalší bádatelia pokúšali zaviesť do výpočtu spolupôsobenie průžkov. Každá z ďalších presnejších metód však stratila jednoduchosť a eleganciu Terzaghiho vzorcov. Naviac vzhľadom k nelinearite skutočného problému, redistribúcii síl a časových faktorov, ako aj presnosti vstupných údajov z prieskumov, mnohí stratili dôveru k zlepšovaniu výpočtových metód.

Zlepšené riešenia, ktoré zavádzajú do výpočtu zložky spolupôsobiacich síl medzi průžkami, sú už staticky neurčitými úlohami, pri ktorých si treba vypomáhať rôznymi odhadmi. Všetky tieto metódy je väčšinou nutné riešiť iteračne a tak plné uplatnenie dosiahli až po zavedení dostupných počítačov.

K metódam, ktoré uplatňujú vzájomnú interakciu průžkov, spomenieme aspoň ich autorov : Krey (1932), Bishop (1954), Breth (1956), Morgenstern a Price (1965), Franke a Spencer (1967). Ďalším vývojovým stupňom riešenia sú špeciálne nelineárne prvky MKP, ktoré už posúvajú problém do matematických výšin (L.Mejzlík, 1975, 1984). Ich nevýhodou je komplikovanosť a zložitejšie definovanie vstupných údajov, správna interpretácia výsledkov a v neposlednom rade cena programov.

Niektoré základné príčiny, ktoré vedú k zosuvom zemného telesa, sú: ● zárez v päte svahu, napr. rozšírenie cesty, uvoľnenie dispozície staveniska ● výkop pri päte svahu pod jej úrovňou, napr. pre drenáž alebo odvodnenie ● zaťaženie v korune svahu, napr. stavebné stroje, násypy ● vibračné účinky dopravy napr. pri kyprých zeminách ● povrchová erózia nechránených svahov (mráz, vietor) ● premočenie svahu v dôsledku dlhotrvajúcich lejakov ● príbojová erózia pri nechránených povrchoch hrádzí ● rýchly pokles hladiny vody v zemných hrádzach, resp. pri povodniach ● napätá hladina spodnej vody v piesčitých medzivrstvách ● tlak vodného stĺpca v erózných zárezoch koruny svahu ● zemetrasenie ● zmena sklonu svahu ● zväčšenie výšky svahu ● zmena vegetácie a následná zmena vodného režimu ● výskyt zlomov a diskontinuít.

V ďalšom sa pokúsime načrtnúť jeden z možných postupov algoritmizácie a niektoré problémy s tým súvisiace k jednej z metód, ktorých podstatu definoval A.W.Bishop, upravený K.Ishiharom (1985) na seizmické účinky. Ako základ algoritmu uvedieme vzorec v zjednodušenej forme (podrobný popis indexácie by bol oveľa rozsiahlejší) :

$$\eta = \frac{\sum((G - Gs)\, tg\varphi + c \,.\, b) / (\cos\alpha + tg\varphi \sin\alpha / \eta))}{\sum(G + Gs + Gw) \sin\alpha + Hs + \sum P.a / R + \sum b.W_A .d / (R \cos\gamma)}$$

kde $Hs = a_g / (g\,.R) \iint r(x,y) \cos\beta\,(x,y)\, Gc_k \, dx\, dy$

Význam znakov je nasledovný :

η - súčiniteľ spoľahlivosti svahu
G - tiaž průžku
Gs - podiel zvislej seizmickej sily
φ - uhol vnútorného trenia zeminy
α - uhol šmykovej plochy s horizontálou
c - súdržnosť
b - šírka průžku
Gw - podiel tiaže podzemnej vody v průžku
Hs - podiel vodorovnej seizmickej sily
P - všeobecné vonkajšie zaťaženie
a - rameno P vzhľadom na stred kružnice šmykovej plochy
R - polomer kružnice šmykovej plochy
W_A - tlak vodného stĺpca na povrch terénu

* Ing.Vladimír Budinský, Aut.Ing., Banská Bystrica

d - rameno výslednice W_A vzhľadom na stred kružnice šmykovej plochy

γ - uhol povrchu terénu s horizontálou

a_g - návrhové seizmické zrýchlenie upravené príslušnými koeficientami

g - gravitačná konštanta

r(x,y) - sprievodič elementárnej hmoty ako funkcia polohy

β(x,y) - uhol sprievodiča s vertikálou

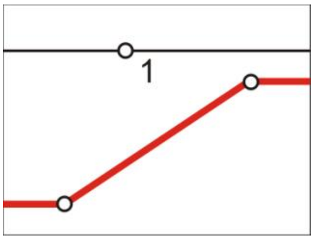
Gc_k - objemová tiaž zeminy a vody započítateľná pre seizmicitu

Pozorný čitateľ si určite všimol, že η, ktoré chceme vypočítať, sa nachádza aj na pravej strane výrazu. Prehoďme ho teda na ľavú stranu a vyextrahujme čisté η. Takto jednoducho to ale nepôjde, lebo η na pravej strane je pod sumáciou a to je problém. Budeme musieť preto η na pravej strane vhodne zvoliť a iteráciou sa priblížiť k presnému riešeniu.

Na začiatku si stanovíme všeobecné požiadavky, ktoré by mal taký program riešiť :

● ľubovoľné zadávanie tvaru terénu a geologických vrstiev ● všeobecný tvar a počet tuhých telies (oporné múry, norné steny a pod. ● krivka hladiny vody (spodnej aj voľnej hladiny) ● ľubovoľný počet zaťažení na terén ● zemné kotvy vo forme zadania vektora sily ● možnosť zadania predpísaného tvaru šmykovej plochy ako alternatívy k valcovej ploche ● stupeň seizmicity územia vo forme a_g ● jednoduchosť programu ● prívetivosť k užívateľovi.

Filozofia jednoduchosti programu : Užívateľ môže veľmi rýchle riešiť jednoduché úlohy. Najjednoduchšie spustenie programu ilustruje **obrázok 1** : ● 1x dvojklik myšou - otvorenie programu ● 4x kliknutie - zadanie tvaru terénu ● 3x kliknutie - zadanie geologickej vrstvy ● 1x kliknutie - spustenie výpočtu.



Obrázok 1

Výpočet samozrejme prebehne s prednadstavenými vlastnosťami vrstvy. K praktickej funkčnosti programu treba zrejme upresniť parametre geologickej vrstvy : merný tiaž, uhol vnútorného trenia a súdržnosť. Tým je program úplne funkčný na jednoduchú úlohu. Avšak vyriešením elementárnych úloh na univerzálnej platforme môžeme v riešení postupne nabaľovať funkcie zovšeobecnelé až do takej úrovne, že v praxi takú zložitosť ani nevyužijeme.

Pri algoritmizácii stability svahu tvoria asi 80 % programovej práce problémy matematickej geometrie a len 20 % geotechnické problémy. Týchto spolu 100 % zase tvorí len asi 20 % tematických problémov a 80 % práce zaberá vzťah užívateľ - počítač. Pri použití objektovo orientovaných jazykov (napr. pod Windows) sa tieto percentá vyrovnávajú v prospech riešenia tematických úloh, nároky na programátora však neklesajú, skôr naopak.

Pri programovaní interaktívnych vstupov všeobecného charakteru musí programátor riešiť aj úlohu, ako užívateľ nesmie zadať vstup, resp. kontrolu vstupných údajov. Táto fáza môže zaberať 50 ÷ 200 % času z programovania vlastných vstupov. Ako príklad môže slúžiť úloha v závere článku.

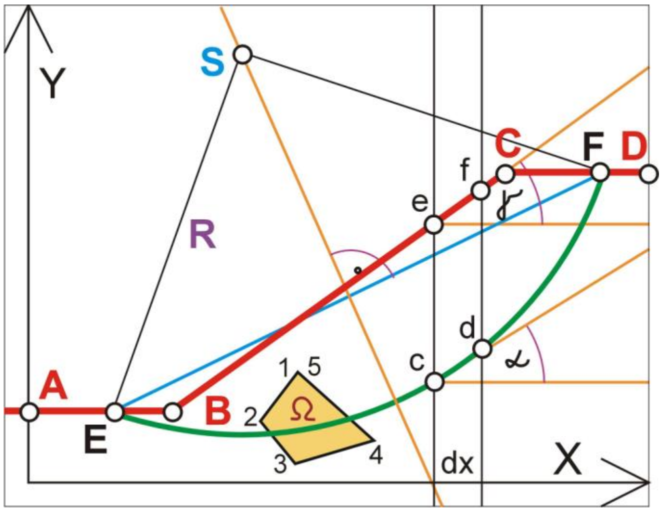
Základnou geometrickou úlohou je pretínanie dvoch kriviek, t.j. určenie súradníc ich priesečníka a faktu, či sa krivky vôbec pretínajú. Našťastie máme vec zjednodušenú tým, že ide o všeobecne rovinné úlohy.

Majme dve krivky zadané explicitnými rovnicami $y = f(x)$ a $y = g(x)$. Priesečník má súradnicu x, ktorá spĺňa rovnicu $f(x) = g(x)$. Je to obecná nelineárna rovnica pre neznámu x a jej všeobecné riešenie je prakticky nerealizovateľné. Už len pri riešení obyčajnej kvadratickej rovnice by sme mali problém, ktorý koreň použiť. Preto tento spôsob využijeme len pre triviálnu úlohu priesečníka dvoch priamok o rovniciach $y = ax + b$. Malé problémy už nastanú, pokiaľ riešime priesečník úsečky a priamky, alebo dvoch úsečiek. A to je pre nás veľmi frekventovaná úloha, ktorá sa však dá pomerne ľahko zdolať.

Ale čo s krivkami? Aktuálnou krivkou je pre nás kružnica, resp. oblúk ako časť kružnice. Všetky ostatné krivky nahradíme polygónmi a tým zredukujeme základnú úlohu na problém najviac priesečníka kružnice a priamky, resp. úsečky a oblúka. Vždy, keď sa dostaneme do situácie, že počítač nad nami vyhráva (je to bohužiaľ často, a najmenej 1:0), treba ísť naňho fintou, úskokom, fígľom. Počítač je hlúpy, úskokom nerozumie a to musíme vždy využiť. Všeobecne dvojparametrická analytická úloha, numericky neriešiteľná, sa voľbou jedného parametra mení na jednoparametrickú a tým numericky ľahšie zvládnuteľnú, pričom voľba prvého parametra prechádza celou svojou množinou, zníženou deterministicky na množinu konečných prvkov. Aplikáciou tejto pomerne zložitej vety posunieme problém tam, kde ho chceme mať. Počítač pradie spokojnosťou, že konečne má čo rátať (a nemusí si čas krátiť animovanými kancelárskymi sponkami, čo gúľajú očami) a nám sa riešenie dostáva pomaly do zorného uhla.

Niekedy je vhodnejšie (časovo) algoritmus vynájsť, ako ho hľadať v literatúre alebo dostupných databázach. Zovšeobecnenie tejto vety by však mohlo viesť k tomu, že na riešenie nemusíme prísť do konca života.

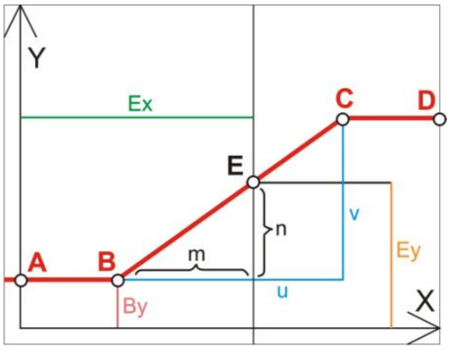
Obrázok 2



Na **obrázku 2** vidíme základný jednoduchý prípad usporiadania geometrických prvkov (geotechnicky valcovú šmykovú plochu s vyznačením povrchu terénu hrubou červenou čiarou). Známe sú pritom len súradnice terénu ABCD a súradnice vrcholov polygónu Ω 1,2,3,4,5, kde 1≡5. Potrebujeme vypočítať najmenej :

● priesečníky šmykovej kruhovej plochy s terénom E,F ● súradnice x,y bodov c,d,e,f ● či kruhový výsek pretína teleso Ω ● uhol α a γ ● súradnice stredu S ● polomer R.

Keď vyriešime tieto základné veci, môžeme si dovoliť prehlásiť, že sa pohneme ďalej. Ako však vidíme, máme oveľa viacej neznámych, ako by nám bolo milé, inými slovami, ako podmienok jednoznačného riešenia. Nevieme vôbec nič o kružnici, kde by mala ležať a aká by mala byť veľká. Keď riešime problém na rysovacej doske, je to pomerne jednoduché. Máme predsa predstavu, ako by mala asi vyzerať šmyková plocha, kam sa bude svah zosúvať a pod. Počítač však zatiaľ nevie úplne, ale úplne nič (predpokladáme, že sa od svojho zapnutia prebojoval aspoň k úrovni poznania vyššieho jazyka). Teda nielen to, kde by mal kruhový výsek umiestniť, on ani nevie, kde je hore a kde dole. Použijeme preto vyššie spomenutý postup a budeme si pomáhať tými „fintami“.



Obrázok 3

Najprv zvolíme x-ové súradnice bodov E a F, ktoré budú tvoriť tetivu kružnice. Keďže zatiaľ nevieme, kde by sme ich mali zvoliť, zvolíme ich všetky. Všetky možné kombinácie. Množinu si však zúžime, aby sme sa celkom nezbláznili (teda hlavne počítač) a to takto :

● sprava ohraničíme množinu pravým okrajom obrazovky, resp. oblasti riešenia, zľava nulou ● F(x) > E (x) ● F(x) > E (x) + L / 5 kde za L sme zvolili maximálnu x-ovú súradnicu, teda šírku oblasti ● interval dx bude napr. L / 100 ● E(x) < (L - L / 5).

Týmito podmienkami sme veľmi jednoducho vymedzili možné základné polohy tetív kružníc. Ostáva nám vypočítať y-ové súradnice E a F. Najprv vypočítame E. Podľa značenia na **obrázku 3** platí :

$$Ey = By + n \qquad \frac{n}{m} = \frac{v}{u} \qquad n = \frac{v \,.\, m}{u} = \frac{(Cy - By) \,.\, (Ex - Bx)}{(Cx - Bx)}$$

$$Ey = By + \frac{(Cy - By) \,.\, (Ex - Bx)}{(Cx - Bx)}$$

Zápisom v didakticky orientovanom jazyku PASCAL by celý algoritmus pre všeobecné riešenie znel ako na **obrázku 4**. Keď preložíme tento obrázok do ľudskej reči, mohlo by to vyzerať asi takto :

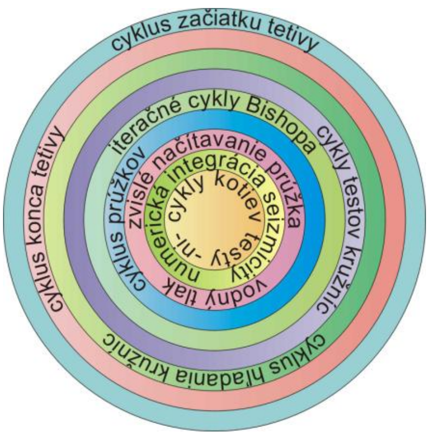
„Predstavujeme vám premenné ***i*** a ***pocetBodovTeren*** ako celé čísla, ***Ex*** a ***Ey*** ako reálne čísla a ***teren*** je množina súradníc x,y ako podmnožina reálnych čísel. Začni opakovať nasledujúci výpočet toľkokrát, ako je v množine terénu bodov, avšak o jeden menej. Ak ***Ex*** je väčšie ako ***i***-tá x-ová súradnica terénu a zároveň menšie ako o jednu väčšia x-ová súradnica terénu, môžeš vypočítať ***Ey*** podľa uvedeného vzorca. Nezabudni skončiť.“

```
var
    i, pocetBodovTeren : integer ;
    Ex, Ey : real ;
    teren : array [1..2,1..maxBodov] of real ;
begin
for  i:=1  to  pocetBodovTeren -1  do
 if Ex>=teren[1, i ]  and  Ex<teren[1,i+1]  then
   Ey := teren[2,1] + (teren[2,i+1] - teren[2, i ])
 * (Ex - teren[1, i ]) / (teren[1, i+1] - teren[1, i ]) ;
end ;
```

Obrázok 4

Rovnaký algoritmus použijeme pre body F,e,f. Výpočet y-ovej súradnice bodov „c“ a „d“ už bude o stupeň ťažší, lebo ich polohu neudáva priamka polygónu, ale kružnica, ktorú ešte ani nemáme. Máme však už jej tetivu EF. Všetky stredy kružníc budú ležať na kolmici k tetive, vedenej od jej stredu. A to je už jednoduchá geometria, úloha pre pána Pytagora. Polomer kružnice môže byť rôzny, opäť nevieme aký, pre istotu zvolíme zase všetky kružnice s týmito obmedzeniami :

Obrázok 5



● stred bude nad priesečníkom tetivy a kolmice ● prvú vzdialenosť stredu S od priesečníka zvolíme napr. ako dĺžku tetivy EF / 8 ● každá ďalšia vzdialenosť bude 2x väčšia ako predchádzajúca ● zvolíme 5 krokov voľby stredu S.

Keby sme si tento postup s piatimi kružnicami narysovali, uvideli by sme veľmi vyrovnané odstupňovanie polomerov kružníc a ich výsekov ako našich už budúcich šmykových plôch. Postup môžeme samozrejme ľubovoľne zahustiť, ale pozor! Aj ten najrýchlejší počítač má svoj strop v spotrebe strojového času a ten ešte budeme potrebovať.

Nie všetky výseky nad tetivou budú pre nás stráviteľné. Medzi ne nebudú patriť napr. tieto :

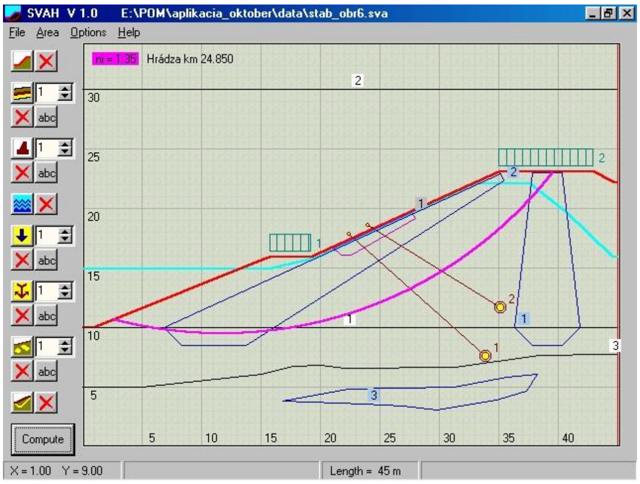
● ktoré vyskočia časťou oblúka z obrazovky ● ktoré vyskočia z povrchu terénu ● ktorých časť oblúka bude mať x-ovú súradnicu väčšiu ako bod F ● ktoré budú pretínať niektoré tuhé teleso (napr. oporný múr, na obrázku 2 teleso Ω).

Pre všetky tieto prípady zostavíme podmieňovacie príkazy alebo podprogramy, a tieto oblúky vyradíme z ďalšieho riešenia.

Akokoľvek elegantne a jednoducho vyzerá vyriešenie tvaru a umiestnenia šmykových plôch, **obrázok 5** ilustruje schému zacyklenia algoritmov hlavného riešiča, čo už celkom jednoduché nie je, naviac to musí aj fungovať.

Keby sme uvažovali, že procesor zvládne úlohy jedného cyklu za jednu milióntinu sekundy, výpočet banálneho problému stability by

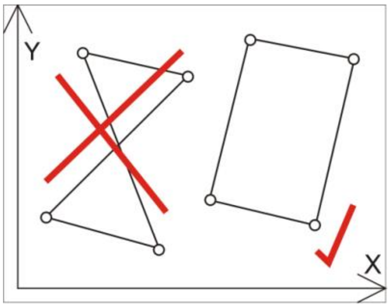
Obrázok 6



sme museli počítať na dni. Uplatnením ďalších „úskokov“ do riešenia cyklov vieme stlačiť výpočet do niekoľkých sekúnd. Medzi ne patrí napr. :

● okamžité vybehnutie z cyklu, pokiaľ sa splní vyžadovaná podmienka ● predradenie jednoduchých cyklov pred časovo náročnejšie, pričom jednoduchý cyklus otestuje, či je zložitejší vôbec potrebný ● používanie jednoduchších premenných, zaberajúcich menej pamäťového miesta, všade tam, kde sa len dá.

Obrázok 7



Filozofiu jednoduchého ovládania programu ilustruje **obrázok 6**, kde získavame čo najviac priestoru pre grafickú časť a intuitívne ovládanie programu pomocou piktogramov, pričom program by nemal strácať plnokrvnosť riešenia väčšiny problémov stability svahu.

ZÁVER

Skúste si na domácu úlohu vypracovať algoritmus na test všeobecného uzavretého n-polygónu tak, aby sa vzájomne nepretínala žiadna z jeho strán (**obrázok 7**). Nie je to až také jednoduché. Kritériá pre splnenie úlohy sú jednoznačné : Musí správne pracovať. Po úspešnom zvládnutí tejto úlohy máte najmenej jednu tisícinu programu za sebou.

LITERATÚRA

[1] Quido Záruba, Vojtěch Mencl : Sesuvy a zabezpečování svahů, Academia 1987

[2] Jozef Hulla, Peter Turček : Zakladanie stavieb, JAGA 1998

[3] Ladislav Drs : Plochy ve výpočetní technice, SNTL 1984

[4] Vladimír Budinský : Manuál k programu SVAH V 1.0, 2002

[5] Leoš Hobst, Josef Zajíc : Anchoring in Rock and Soil, Elsevier Amsterdam, 1983

[6] Alois Myslivec, Jaroslav Eichler, Ján Jesenák : Mechanika zemin, SNTL 1970

[7] STN 73 0036 Seizmické zaťaženie stavebných konštrukcií, 1997

[8] STN P ENV 1998-5 Návrhové požiadavky na seizmickú odolnosť konštrukcií Časť 5: Základy, oporné konštrukcie a geotechnické hľadiská, 1999

Annotations

Algorithmization of the stability of a slope

The problem of the stability of a slope is still actual. This year's floods have prove it. Lots are searched for also on places that were unsuitable in the past. The system of highways needs new routes. The article first provides a simplified introduction to the problem of the stability of slopes and indicates, in a simple form, possibilities of the automation of calculation procedures.