

УДК 621.7:519.85

В. М. Михалевич, д. т. н., проф.; Ю. В. Добранюк; О. В. Михалевич
ПОБУДОВА ЕФЕКТИВНИХ ОБЧИСЛЮВАЛЬНИХ СХЕМ У MARLE ПІД
ЧАС РОЗВ'ЯЗАННЯ ЗАДАЧІ ВИЗНАЧЕННЯ ГРАНИЧНИХ
ДЕФОРМАЦІЙ ЗА УМОВ СКЛАДНОГО ДЕФОРМУВАННЯ

Розглянуто задачу визначення граничних деформацій за умов складного деформування, яке має місце на вільній бічній поверхні під час вісесиметричного осадження циліндричних зразків за різних умов тертя на торцях. Висвітлено ключові моменти цієї задачі. Показано, що застосування стандартних команд системи Maple для розв'язання розглядуваної задачі не є ефективним з точки зору часових витрат на проведення обчислень. Проаналізовано різні підходи до побудови ефективних обчислювальних схем. Виявлено, що найефективнішою є схема, яка базується на застосуванні комбінованого методу хорд та дотичних.

***Ключові слова:** напружено-деформований стан; осадка циліндричних зразків; накопичена деформація, модель руйнування; ітераційні методи.*

Постановка проблеми та аналіз останніх досліджень

Об'єкт дослідження. Процеси складного деформування.

Предмет дослідження. Математична модель граничних деформацій на вільній бічній поверхні під час вісесиметричного осадження циліндричних зразків за різних умов тертя на торцях.

Мета дослідження. Пошук ефективних обчислювальних схем у середовищі математичного пакета Maple для задачі визначення граничних деформацій за умов складного деформування, яке відбувається під час вісесиметричного осадження циліндричних заготовок за різних умов тертя на торцях.

Задачі дослідження: 1. Вибір моделі накопичення пошкоджень. 2. Визначення моделі напружено-деформованого стану на вільній бічній поверхні при вісесиметричному осадженні циліндричних зразків за різних умов тертя на торцях. 3. Постановка математичної задачі визначення граничних деформацій за умов складного деформування та дослідження альтернативних методів її розв'язання в середовищі Maple. 4. Визначення ефективної схеми розв'язання задачі та наочного подання залежності граничних деформацій від процесу зміни напружено-деформованого стану критичної області заготовки.

Актуальність роботи. Розв'язання задач теорії деформівності вимагає значних обчислювальних витрат [1 – 3]. І незважаючи на потужність сучасних обчислювальних засобів, у більшості випадків, задача розробки ефективних обчислювальних схем залишається доволі актуальною.

Основна частина

Згідно класифікації, яку запропоновано в [1], складним деформуванням будемо називати такий процес немонотонного деформування, при якому компоненти напрямного тензора приростів пластичних деформацій змінюються неперервно. Для визначення граничних деформацій при складному деформуванні в рамках теорії неперервних пошкоджень розроблено ряд моделей [1, 2, 4, 5]. Розглянемо найпростіший варіант: модель, яка базується на лінійному принципі накопичення пошкоджень – модель Колмогорова [4, 2, 1]:

$$\psi(\varepsilon_u) = \int_0^{\varepsilon_u} \frac{d\varepsilon_u}{\varepsilon_{*c} \cdot [\eta(\varepsilon_u)]}, \quad (1)$$

де ψ – пошкодженість макрочастинки, яка змінюється від 0 у початковому стані до 1 при

досягненні граничного стану; ε_u – накопичена пластична деформація:

$$\varepsilon_u(t) = \int_0^t \dot{\varepsilon}_u(\tau) d\tau, \quad (2)$$

де τ, t – час; $\dot{\varepsilon}_u$ – інтенсивність швидкостей деформацій; $\varepsilon_{*c} = \varepsilon_{*c}(\eta)$ – крива граничних деформацій за умови стаціонарного деформування; η – показник напруженого стану [1, 2, 6].

З урахуванням (1) умова досягнення граничного стану набуває вигляду:

$$\psi(\varepsilon_*) = 1, \quad (3)$$

або

$$\int_0^{\varepsilon_*} \frac{d\varepsilon_u}{\varepsilon_{*c} \cdot [\eta(\varepsilon_u)]} = 1, \quad (4)$$

де ε_* – гранична деформація за умов немонотонного, зокрема, складного деформування.

Як видно із зображення (4), знаходження граничної деформації ε_* зводиться до розв'язання нелінійного рівняння. Нелінійна функція подана визначеним інтегралом зі змінною верхньою межею. Слід зазначити, що зовнішньо просте рівняння (4) в окремих випадках може призвести до громіздких співвідношень і, головне, до значних часових витрат для отримання розв'язку.

Для визначення напружено-деформованого стану на вільній бічній поверхні при вісесиметричному осадженні циліндричних заготовок скористаємося результатами, які наведені в роботах [7, 8]. Криву граничних деформацій за умови стаціонарного деформування апроксимуватимемо виразом:

$$\varepsilon_{*c}(\eta) = \varepsilon_{*c}(\eta = 0) \cdot \left(\frac{(1 - \eta) \cdot \varepsilon_{*c}(\eta = -1)}{2 \cdot \varepsilon_{*c}(\eta = 0)} + \frac{(1 + \eta) \cdot \varepsilon_{*c}(\eta = 0)}{2 \cdot \varepsilon_{*c}(\eta = 1)} \right)^{-\eta}, \quad (5)$$

де $\varepsilon_{*c}(\eta = -1), \varepsilon_{*c}(\eta = 0), \varepsilon_{*c}(\eta = 1)$ – гранична деформація за умов стиску, кручення та розтягу відповідно.

З урахуванням числових значень параметрів моделі, рівняння (4) може бути подано у вигляді: $\left(\alpha \in \left[0, \frac{\pi}{2} \right], m > 0 \right)$

$$m \cdot \int_0^\alpha \frac{1,47 \cdot \sqrt{3 + \frac{1}{\cos^4 x}}}{\left(\frac{(1 - \cos^2 x) \ln \left(\frac{0,94(1 - \cos^2 x)}{\sqrt{1 + 3 \cos^4 x}} \right)}{\sqrt{1 + 3 \cos^4 x}} \right)} dx - 1 = 0, \quad (6)$$

де m – відображує умови тертя на торцях та вважається сталою у процесі дослідження конкретного зразка $[0, 0]$; α – параметр процесу осаджування.

Корінь рівняння (6) позначатимемо через α_* , $\alpha_* \in \left[0, \frac{\pi}{2} \right]$.

Зауважимо, що співвідношення (5), (6) наводяться вперше в цій роботі. Викладенню методики розв'язання цих рівнянь буде присвячено окремі роботи.

Отже, задача визначення граничних деформацій за умов вісесиметричного осаджування циліндричних зразків за різних умов тертя на торцях зводиться до розв'язання рівняння (6) для конкретних значень m . Після знаходження відповідних значень α_* легко може бути знайдено координати граничної точки на відповідному шляху деформування

$\eta(\varepsilon_u) = \eta[\varepsilon_u(\alpha)]$. У цій роботі розглянемо такі альтернативні способи розв'язання поставленої задачі: 1. Безпосереднє застосування стандартної команди Maple системи для отримання чисельного розв'язку нелінійного рівняння; 2. Розвинення підінтегральної функції в степеневий ряд з подальшим інтегруванням, що повинно підвищити швидкість обчислення значень нелінійної функції, яка досліджується. 3. Застосування ітераційних методів чисельного знаходження розв'язку нелінійного рівняння.

У системі Maple для чисельного розв'язання одного рівняння або системи рівнянь (лінійних або нелінійних) передбачено команду `fsolve`.

Для обчислення значень лівої частини рівняння (6) за конкретних значеннях m , γ створимо процедуру `PsK(m, α)`:

```
> PsK :=
(m, alpha) -> evalf(m*Int(1.47*sqrt(3+1/(cos(x)^4)/exp(...)), x=0..
alpha)-1);
```

З метою економії місця та запобігання розсіюванню уваги очевидна частина програмного коду опущена. Фактично ця процедура реалізує чисельне знаходження визначеного інтеграла за допомогою стандартної команди `evalf`. Принциповим у процедурі є те, що команда знаходження інтеграла `Int` записана з великої букви в інертній формі. У такому випадку система зразу ж застосовуватиме чисельний алгоритм знаходження інтеграла. А якщо ні, то під час задання інтеграла командою `int`, система спочатку намагатиметься знайти інтеграл у символічному вигляді, що забиратиме певний час, і тільки після того, як упевниться, що первісної знайти не може, автоматично підключить алгоритм чисельного інтегрування.

Для застосування команди `fsolve` також створимо процедуру `Розв_к`:

```
> `Розв_к` := proc(m, `діапазон` :: range)
if type(m, numeric) then
fsolve(PsK(m, t)-1, t, `діапазон`)
else
`procname`(m, `діапазон`)
end if
end proc;
```

Слід зазначити, що наявність у процедурі `Розв_к` конструкції з оператором `procname`(m , `діапазон`) для обробки ситуації у випадку нечислового значення параметра m , є ключовою для побудови графіка функції $\alpha_* = \alpha_*(m)$.

Обчислення кореня для конкретного значення m система виконує приблизно за 7 сек., що легко визначити за допомогою конструкції:

```
> `Старт` := time(): `Розв_к`(0.09, 0..1.5); `Фініш` := time() -
`Старт`;
1.380775711
T_k 7.344
```

На побудову графіка $\alpha_* = \alpha_*(m)$ в діапазоні $m \in [0, 0.1, 8]$ система витрачає більше 20 хвилин. Звісно, в сучасних умовах така ситуація неприйнятна, оскільки, зокрема, час розв'язання не корелюється із ступенем складності задачі. Всі обчислення проводили на комп'ютері з центральним процесором AMD Sempron, 1500 MHz (9 x 167) 2200+.

Підхід, якій базується на розвитку підінтегральної функції в степеневий ряд з подальшим інтегруванням, виглядає заманливим у зв'язку з легкістю реалізації цих операцій у середовищі системи Maple. Передбачалося, що заміна поліномом складної функції, яка характеризується визначенням інтегралом із змінною верхньою межею, повинна прискорити чисельне знаходження кореня нелінійного рівняння.

За допомогою команд `taylor(f(x), x=0, 17)`; `convert(%polynom), x)`, де $f(x)$ – підінтегральна функція в співвідношенні (6), отримаємо відповідний поліном:

$$0,77 + 1,11 \cdot x^2 + 1,38 \cdot x^4 + 1,32 \cdot x^6 + 1,0 \cdot x^8 + 0,59 \cdot x^{10} + 0,23 \cdot x^{12} + 0,01 \cdot x^{14} - 0,07 \cdot x^{16}. \quad (7)$$

На рис.1 подано співставлення графіків підінтегральної функції $f(x)$ та її розвинення в ряд (7) в околі точки $x=0$:

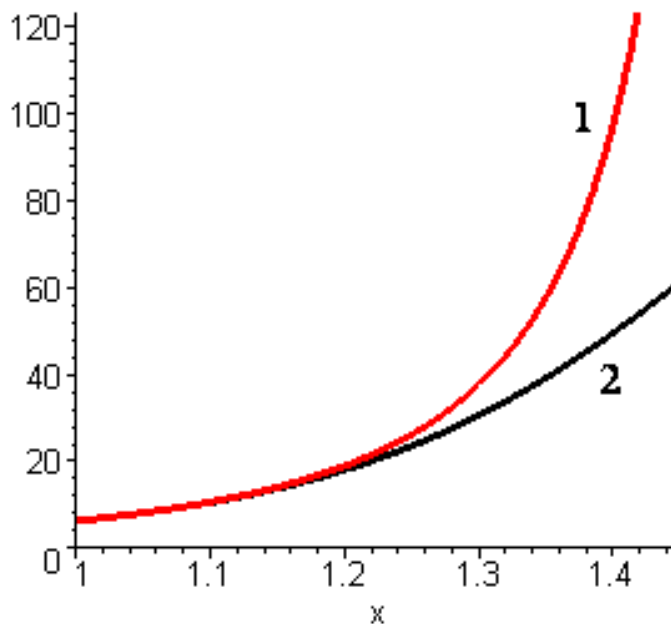


Рис. 1. Співставлення графіків підінтегральної функції $f(x)$ та її розвинення в ряд (7) в околі точки $x=0$, де
1 – функція $f(x)$; 2 – поліном (7)

Як видно, при $x > 1,18$ узгодження не можна визнати задовільним. Підвищенням степеня полінома та зміною точки, навколо якої відбувається розкладання, суттєвого покращення результату досягти не вдалося. Під час використання апроксимації (7) корені рівняння (6) з задовільною точністю отримуються тільки для $m > 0,18$. Отже, в цьому випадку мова вже йде не про швидкість обчислень, а про незадовільну точність отримання результатів.

Ітераційні методи де-факто найбільш поширені методи отримання розв'язків нелінійних рівнянь. До лідерів ітераційних методів відносяться методи хорд, дотичних та січних.

Принципова відмінність методу січних [9] від методу дотичних у тому, що в першому методі на кожному кроці обчислюється тільки нове значення функції, в той час, коли в другому методі на кожному кроці обчислюється як значення функції, так і її похідної. Тому, незважаючи на менш швидку збіжність методу січних порівняно з методом дотичних, за рахунок меншої кількості обчислень на кожному кроці, перший метод може виявитися більш ефективним. Оскільки в розглядуваному випадку функція становить визначений інтеграл зі змінною верхньою межею, то часові витрати на обчислення значень похідної від функції нехтовно малі порівняно із обчисленням значень самої функції. Отже, наведені міркування дозволяють відкинути метод січних та зосередитись на методах хорд та дотичних.

Кожний із цих методів має один і той самий недолік, який полягає в тому, що наближення до кореня відбувається з однієї сторони. Цей факт ускладнює побудову критерію завершення ітерацій.

Застосування комбінованого методу хорд та дотичних дозволяє отримати зручну ознаку досягнення заданої точності: кожний із цих методів породжує послідовність значень, які наближуються до кореня з різних сторін. Обчислення завершуються за умови, що різниця між поточними наближеннями до кореня, які обчислені за двома методами, стане меншою потрібного ступеня точності.

Для застосування методу хорд потрібно виділити відрізок, який утримує корінь та на

якому функція монотонна. Для застосування метода дотичних потрібно виділити відрізок, який утримує корінь та на якому функція не має точок перегину. Можна довести, що функція, яка визначається лівою частиною рівняння (6) монотонно зростаюча та опукла донизу на області визначення за будь-яких значень $m > 0$. Автори знайшли вагоме доведення цього факту, чому буде присвячено окрему роботу. Отже, за методом хорд наближення до кореня відбуватиметься зліва, а за методом дотичних – справа.

Показовою є реалізація в середовищі системи Maple методу дотичних (метод Ньютона) [10]. Приведемо процедуру, яка за заданою функцією автоматично генерує формулу ітерації.

```
> `М_Ньютон` := proc (y : procedure)
    ( (m, x) -> x) - eval (y) / D [2] (eval (y) )
end proc;
```

У цьому випадку легко записати формулу ітерації:

$$\alpha_{n+1} = \alpha_n - \frac{F(m, \alpha_n)}{f(\alpha_n)}, \quad (8)$$

де α_n – поточне наближення кореня α_* співвідношення (6); $F(m, \alpha_n)$ – ліва частина рівності (6); $f(x)$ – підінтегральна функція в співвідношенні (6).

У переважній більшості інших випадків наведена процедура позбавляє необхідності виконання, інколи доволі громіздкої, операції диференціювання.

Команда

```
> `Колм_Ньют` := `М_Ньютон` (PsK) ;
```

дозволяє отримати функцію для обчислення методом дотичних поточного наближеного значення кореня рівняння (6).

Потрібно зауважити, що надмірна складність може привести до помітного погіршення ефективності обчислень. Реалізуємо метод хорд наступними процедурами:

```
> `М_хорд` := proc (F : procedure, m, x1, x2)
    evalf (x1 * F (m, x2) - x2 * F (m, x1) ) / ( F (m, x2) - F (m, x1) ) ;
end proc;
```

Отримуємо конкретний вираз для функції PsK:

```
> `Колм_Хорд` := `hord` (PsK, m, X1, X2) .
```

Перетворюємо вираз у функцію:

```
> `Колм_Хорд` := unapply ( `Колм_Хорд` , m, X1, X2) .
```

Вирази, які виводяться на екран у результаті роботи цих процедур, не наводимо в зв'язку з отриманням занадто громіздких виразів. Відтворивши приведені команди в середовищі Maple, ці вирази можна побачити. Вираз, який використовується в процедурі для обчислення поточних наближень кореня методом хорд містить декілька елементів, які повторюються. Зокрема, вираз:

$$\sqrt{1 + 3 \cdot \cos^4 x}$$

зустрічається тричі і кожного разу обчислюється заново. З точки зору основ програмування така ситуація не витримує ніякої критики. Але це ще раз підтверджує, що коли мова йде про чисельні обчислення, цих основ потрібно дотримуватися, незважаючи на шалений ріст обчислювальної потужності сучасних комп'ютерів. У цьому випадку для досягнення точності 0,005 при $m=2$ комбінованим методом знадобилося виконати 80 кроків, а вже для досягнення точності 0,0005 знадобилося 788 кроків. Причому аналіз результатів показав, що така неприродно велика для комбінованого методу кількість ітерацій пов'язана саме з поточними наближеннями, які обчислюються за методом хорд. У той час, коли метод дотичних уже на четвертій ітерації давав результат, який перевищував задану точність та залишався незмінним на всіх наступних ітераціях. Чому саме це відбувається автори не стали пояснювати, внаслідок того, що саму ситуацію було створено, у цьому випадку, штучно.

Для побудови графіка кривої граничних деформацій при осадці циліндричних зразків за різних умов тертя на торцях створимо наступну процедуру:

```

> `Графік` :=proc (m, X1, X2, dx)
local a, b, x0;
if type(m, numeric) then
  a:=X1:b:=X2:x0:=X2:i:=0:
  while (b-a)>=dx do
    x0:=`Колм_Ньют`(m, x0);
    b:=x0;
    a:=`hord`(PsK, m, a, b);
  end do:
  (a+b)/2
else
  'procname'(m, X1, X2, dx)
end if
end proc:

```

Наведена процедура є ключовою для можливості побудови графіка на рис. 2:

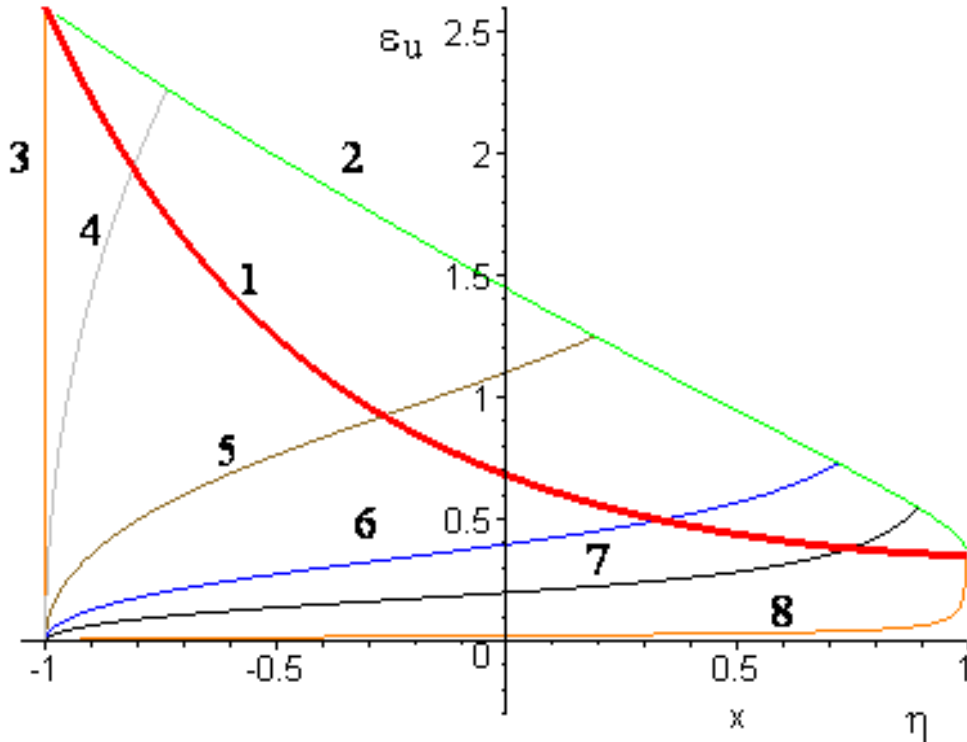


Рис. 2. Моделювання граничних деформацій при вісесиметричному осадженні циліндричних зразків, де: 1 – крива граничних деформацій під час стаціонарного деформування; 2 – крива граничних деформацій під час складного деформування; 3÷8 – криві, які відображають шляхи деформування за різних умовах тертя на торцях (параметр m): 3 – $m = 30$, 4 – $m = 2$, 5 – $m = 0,5$, 6 – $m = 0,18$, 7 – $m = 0,09$, 8 – $m = 0,01$

Тривалість побудови системою даного графіка не перевищує 20 с, що є цілком прийнятним. У той же час, необхідність застосування замість представлення (1) моделей, які базуються на нелінійному законі підсумовування пошкоджень, зокрема, запропонованому в [2], приведе до суттєвого збільшення кількості обчислень. Це пов'язано з тим, що під інтегралом у співвідношенні (6) з'являється величина, яка зображена у вигляді інтеграла. Застосування тензорних моделей [1] приведе до ще більшої кількості обчислень. У результаті може виникнути необхідність вдосконалення схеми, яка побудована в цій роботі. Зауважимо, що криву граничних деформацій за умов складного деформування вперше побудовано в цій роботі. Раніше такі криві будували за допомогою інтерполяції та Наукові праці ВНТУ, 2009, № 2

екстраполяції по 3÷5 точках [2, 8].

Висновки

1. Побудована схема обчислень, яка базується на застосуванні комбінованого методу хорд та дотичних. Порівняно із застосуванням стандартної команди системи Maple це дозволило підвищити швидкість побудови графіка, наведеного на рис. 2, в 60 разів, зменшивши тривалість обчислень з 20 хвилин до 20 с.

2. Розроблена методика придатна не тільки для розглянутого процесу осадження циліндричних зразків, але й для будь-яких процесів, які супроводжуються складним деформуванням.

3. Залучення нелінійних та тензорних моделей накопичення пошкоджень може викликати необхідність вдосконалення побудованої схеми обчислень.

СПИСОК ЛІТЕРАТУРИ

1. Михалевич В. М. Тензорні моделі накопичення пошкоджень. – Вінниця: "УНІВЕРСУМ-Вінниця", 1998. – 195 с.
2. Огородников В. А. Деформируемость и разрушение металлов при пластическом формоизменении. – К.: УМК ВО, 1989. – 152 с.
3. Сивак И. О. Влияние немонотонности нагружения на пластичность при радиальном выдавливании с контурной осадкой // Науковий вісник Національної гірничої академії України. – 2001. – С. 47 – 50.
4. Колмогоров В. Л. Напряжения, деформации, разрушение. – М.: Металлургия, 1970. – 229 с.
5. Сопротивление деформации и пластичность металлов при обработке давлением // Калпин Ю. Г., Перфилов В. И., Петров П. А., Рябов В. А., Филиппов Ю. К. – М.: «МАМИ», 2007. – 118 с.
6. Лебедев А. А., Михалевич В. М. О выборе инвариантов напряженного состояния при решении задач механики материалов // Проблемы прочности. – 2003. – № 3. – С. 5 – 14.
7. Смирнов-Аляев Г. А. Сопротивление материалов пластическому деформированию. Инженерные методы расчета операций пластической обработки материалов. – М.- Л.: Машгиз, 1961. – 463 с.
8. Мишулин А. А. Михалевич В. М. Тензорно-нелинейная модель накопления повреждений // Обработка металлов давлением. – Свердловск. – 1985. – С. 10 – 14.
9. Калиткин Н. Н. Численные методы. — М.: Наука, 1978. – 512 с.
10. Maple 9/ Advanced Programming Guide/M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, P. DeMarco. – Canada. Maplesoft, division of Waterloo Maple Inc. – 2003. – 443 p.

Михалевич Володимир Маркусович – д. т. н., професор, завідувач кафедри прикладної математики.

Добрянюк Юрій Володимирович – аспірант кафедри прикладної математики.

Михалевич Олексій Володимирович – студент групи ІБС-07.

Вінницький національний технічний університет.