



Прийнято 03.12.2025. Прорецензовано 13.12.2025. Опубліковано 30.12.2025.

УДК 621.88:004.94

DOI: 10.31471/1993-9965-2025-2(59)-101-114

МУЛЬТИАГЕНТНА СИСТЕМА АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ РІЗЬБОВИХ З'ЄДНАНЬ НАСОСНИХ ШТАНГ

Копей В. Б.*

Доктор технічних наук, професор

Івано-Франківський національний технічний університет нафти і газу

вул. Карпатська, 15, м. Івано-Франківськ, 76019, Україна

<https://orcid.org/0000-0003-0008-8260>

e-mail: vkopey@gmail.com

Проців О. І.

Студент

Івано-Франківський національний технічний університет нафти і газу

вул. Карпатська, 15, м. Івано-Франківськ, 76019, Україна

<https://orcid.org/0009-0008-5204-5055>

e-mail: oleksandr.protsiv-pmkm241@nung.edu.ua

Пронюк І. В.

Аспірант

Івано-Франківський національний технічний університет нафти і газу

вул. Карпатська, 15, м. Івано-Франківськ, 76019, Україна

<https://orcid.org/0009-0005-1029-5337>

e-mail: pronyuk00@gmail.com

Назаренко С. К.

Аспірант

Івано-Франківський національний технічний університет нафти і газу

вул. Карпатська, 15, м. Івано-Франківськ, 76019, Україна

<https://orcid.org/0009-0001-0787-8655>

e-mail: 1314296@ukr.net

Анотація. Різьбові з'єднання насосних штанг використовуються під час видобування нафти свердловинними штанговими насосними установками та призначені для з'єднання штанг в одну колону. Під час експлуатації різьбових з'єднань насосних штанг виникає чимало проблем, пов'язаних із циклічними навантаженнями та високою концентрацією напружень в різьбовому з'єднанні, що призводить до втомних тріщин, поломок та обриву колони. На міцність з'єднань впливає чимало конструкційних факторів, дослідити які можна методом скінченних елементів. Показано принципи розроблення та використання мультіагентної системи для автоматизованого проєктування різьбових з'єднань насосних штанг. Для створення модулів автоматизації побудови геометричної та осесиметричної скінченно-елементної моделей різьбового з'єднання використано

Запропоноване посилання: Копей, В. Б., Проців, О. І., Пронюк, І. В. & Назаренко, С. К. (2025). Мультіагентна система автоматизованого проєктування різьбових з'єднань насосних штанг. Науковий вісник ІФНТУНГ, 2(59), 101-114. doi: 10.31471/1993-9965-2025-2(59)-101-114

* Відповідальний автор



Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

вільне програмне забезпечення Calculix, Gmsh і Python-пакети Cadquery та Pycalculix. Ці пакети значно спрощують побудову параметричних скінченно-елементних моделей, які придатні до повністю автоматизованої оптимізації конструкції за критеріями напружень і деформацій. З використанням цих модулів та моделі акторів для паралельних обчислень на основі Python-пакету Ray удосконалено мультиагентну систему для параметричної оптимізації конструкції різьбового з'єднання за критерієм мінімальних еквівалентних напружень. Система містить агенти з протилежними оптимізаційними правилами (реалізовано сітковий та стохастичний методи), що дозволяє ефективно шукати мінімуми в багатовимірних просторах. Існує також можливість простого створення агентів з іншими правилами та пошуку їхніх ефективних поєднань. З використанням цієї системи досліджено напружено-деформований стан різьбового з'єднання ШН19 в аварійних умовах відсутності попереднього затягування, що трапляється у разі самовідгвинчування з'єднання, та досліджено вплив радіуса зарізьбової канавки та зовнішнього діаметра муфти на еквівалентні напруження. Код розробленої системи автоматизованого проектування опубліковано на GitHub під вільною ліцензією GPL-3.0.

Ключові слова: різьбове з'єднання, автоматизоване проєктування, мультиагентна система, модель акторів, оптимізація, напруження

Вступ

Різьбові з'єднання насосних штангах використовуються під час видобування нафти свердловинними штанговими насосними установками та призначені для з'єднання штанг в одну колону. Колона насосних штанг передає рух від привода (верстат качалка) до свердловинного плунжерного насоса. Різьба ніпелів для насосних штанг виготовляється шляхом накачування, що забезпечує залишкові напруження стиску в западинах витків і високу циклічну довговічність. Проте під час експлуатації різьбових з'єднань насосних штанг, як і багатьох інших різьбових з'єднань, виникає чимало проблем, пов'язаних з циклічними навантаженнями та дуже високою концентрацією напружень в різьбі [1-6]. Частина з цих проблем спричинена також технологічними чинниками [7]. Усе це призводить до втомних тріщин, зношування та руйнування з'єднання [1, 2, 3]. Втома ускладнюється також процесами корозії, зношування різьби та зовнішньої поверхні муфти [8], а також можливістю самовідгвинчування різьбових з'єднань [1, 2, 3]. Оскільки на міцність з'єднання впливає велика кількість параметрів [2], виникає потреба у розробленні параметричних чисельних моделей та ефективного програмного забезпечення для глибокого аналізу та багатовимірної оптимізації конструкції». Критерієм такої оптимізації можуть бути еквівалентні напруження, що виникають в різьбовому з'єднанні.

Аналіз досліджень і публікацій

Відомо чимало досліджень, пов'язаних з удосконаленням та оптимізацією конструкції різьбового з'єднання насосних штанг та інших подібних різьбових з'єднань [2, 3, 5, 6, 9-14]. Часто ці дослідження базуються на скінченно-елементному аналізі (FEA), реалізованому, як правило, в комерційному програмному забезпеченні. Наприклад, за допомогою FEA про-

понували збільшувати довжину зарізьбової канавки [2], або виготовляти її з еліптичним профілем [13]. Це дозволяє збільшити її податливість, значно зменшити концентрацію напружень в зонах перших витків різьби та підвищити циклічну довговічність з'єднання. Відомі також дослідження, пов'язані зі зміною профілю різьби [9]. Така зміна може значно вирівняти навантаження вздовж різьби, але є трудомісткою для впровадження. Інші дослідження [2] пропонують застосування муфти розтягу-стиску, що зменшує напруження на перших витках різьби. В працях [12, 13] за допомогою FEA запропоновано конструкції двоопорного з'єднання з підвищеною жорсткістю. Також можна змінювати інші параметри різьбового з'єднання – збільшувати фаску на муфті, змінювати радіус зарізьбової канавки, застосовувати нерівномірний або неоднаковий крок різьби [2], наносити пластичне покриття на різьбу, виконувати попереднє пластичне деформування різьби високим крутним моментом згвинчування або високим навантаженням розтягу [14].

Формулювання проблеми та цілей дослідження

Таким чином, існує багато параметрів, які можуть впливати на статичну і циклічну міцність з'єднання. Часто необхідно змінювати їх одночасно [2]. Виникає проблема багатовимірної оптимізації конструкції. Більшість відомих скінченно-елементних моделей різьбових з'єднань [9–13] не дозволяють повністю автоматизувати цикл оптимізації, який охоплює зміну параметрів, перебудову моделі, симуляцію та отримання результатів. Такі моделі можна розробити за допомогою інтерфейсів прикладного програмування (API) відомих CAD/CAE-систем. Наприклад, API ABAQUS/CAE та SolidWorks були використані в дослідженнях [2, 13]. Проте це програмне забезпечення є ко-

мерційним і доволі дорогим, що обмежує поширення і розвиток моделей. Існують вільні альтернативи, але вони не є настільки зручними у використанні та потребують додаткових зусиль користувача, глибшого знання математичних основ FEA і програмування. Зокрема, це вільна FEA-система Calculix [15] та вільний генератор скінченно-елементної сітки Gmsh [16]. Для мови програмування Python були розроблені модулі, які дозволяють зручно працювати з цими системами [17]. Також для Python існують вільні пакети для побудови геометричної моделі. Найбільш якісні (Gmsh, FreeCAD, Cadquery), що базуються на вільному геометричному ядрі OpenCascade Technology (OCCT) [18], що підтримує технологію граничного подання BREP. Cadquery [19] цікавий тим, що за допомогою компактного програмного коду дозволяє просто створювати доволі складні геометричні об'єкти, як це робиться у OpenSCAD, але орієнтований на Python та має більше можливостей завдяки BREP.

Задача багатопараметричної оптимізації може бути ускладнена наявністю багатьох локальних мінімумів. Якщо застосовувати тільки алгоритм градієнтного спуску, то існує ризик помилково потрапити в локальний мінімум замість глобального. Якщо ж застосовувати тільки сітковий метод, то результат буде неточний, а обчислення будуть занадто тривалими. Класичні методи оптимізації, такі як пошук по сітці (відомий ще як метод "грубої сили") або градієнтний спуск, не підходять для багатовимірної оптимізації. У різьбових з'єднаннях може бути дуже багато параметрів (10 і більше), які одночасно потребують оптимізації. У цих випадках потрібно застосовувати комбіновані методи оптимізації, які, наприклад, поєднують пошук по сітці з градієнтним спуском або зі стохастичним пошуком. До таких методів можна віднести метаевристичні алгоритми [20]. Зокрема, це стосується використання таких метаевристичних методів, як алгоритми імітації відпалу, диференціальної еволюції, мурашиної колонії та інших. Ці методи намагаються імітувати природні процеси, що стали ефективними завдяки дії екстремальних фізичних принципів чи механізмів природного відбору.

Перспективним методом синтезу таких алгоритмів є мультиагентний підхід [21, 22]. В мультиагентній програмній системі кожний агент – це об'єкт який володіє поведінкою, що нагадує поведінку живої істоти. Він має наміри, переконання та особливості поведінки [21]. Система різнотипних агентів часто є ефектив-

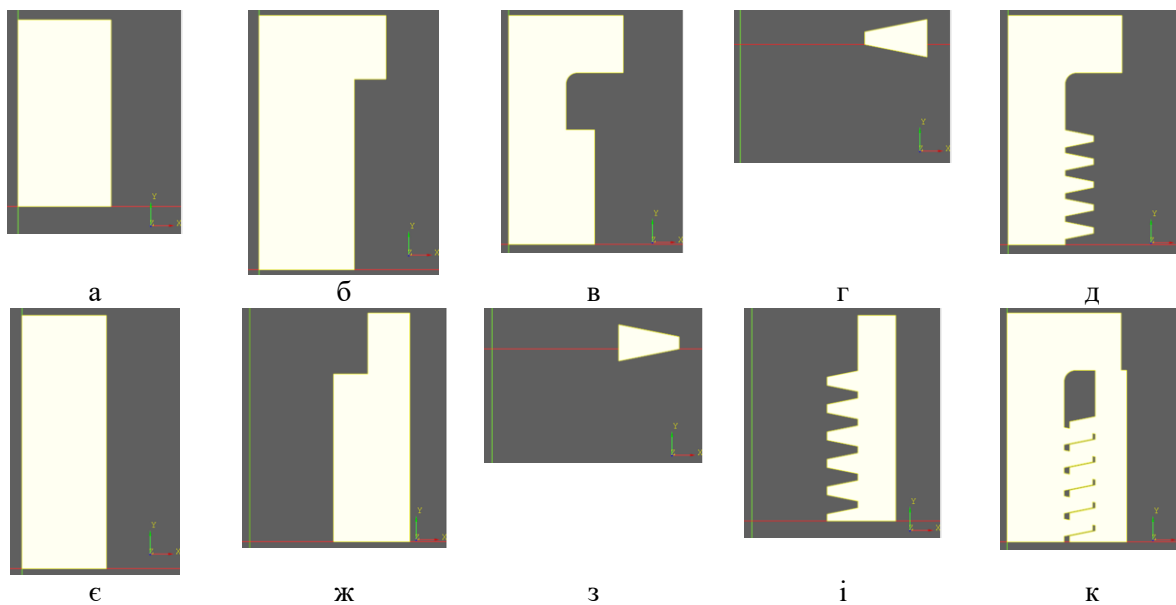
нішою, аніж один агент, що пояснюється системним ефектом. Для оптимізаційних задач можна розробити мультиагентну систему з різнотипними агентами, які ефективно шукають мінімум у багатовимірному просторі з використанням різних правил поведінки [22]. Крім того, така система здатна самовдосконалюватися, орієнтуючись на найкращі стратегії оптимізації.

Проблема оптимізаційного пошуку в багатовимірному просторі ускладнена низькою швидкістю алгоритмів. Для підвищення продуктивності алгоритмів необхідне їх паралельне виконання на багатопроцесорних системах або обчислювальних кластерах. Класичний Python-код не володіє здатністю до простого розпаралелювання. Проте розробники пакету Ray [23] вирішили цю проблему. Пакет Ray використовує концепцію акторів для паралельного виконання завдань і дозволяє просто перетворити будь-який Python-код в паралельну версію. Зокрема Ray Tune з успіхом застосовується в задачах оптимізації гіперпараметрів моделей машинного навчання та дозволяє паралельно запускати різні конфігурації оптимізації, які досліджують простір рішень. Це відповідає ідеї розподілених оптимізаційних агентів.

Метою дослідження є створення принципів розроблення та використання мультиагентної системи для автоматизованого проектування муфтових різьбових з'єднань насосних штанг з використанням вільного програмного забезпечення.

Висвітлення основного матеріалу дослідження

Розроблений Python-модуль `run_cq2.py` [24] використовує пакет Cadquery для побудови параметричної моделі різьбового з'єднання за стандартом ГОСТ 13877-96. Для інтерактивної візуалізації моделі під час її побудови використано програму CQ-editor [19]. Проте вона є не обов'язковою для роботи пакету, а тільки полегшує створення складних моделей та відлагодження програми. Також вона рекомендована для вивчення пакету Cadquery. Спочатку створено циліндричну заготовку ніпеля без вказання реальних розмірів (рис. 1а). На площині XY їй відповідає прямокутник, ліва сторона якого збігається з віссю OY. Операції (вирізання, об'єднання, скруглення та інші) виконуються над тривимірними формами, тому реально необхідно створити не прямокутник, а прямокутний паралелепіпед з довільною висотою по осі Z. Далі шляхом операції вирі-



а – заготовка ніпеля; б – розточування ніпеля; в – зарізьбова канавка ніпеля; г – профіль різьби ніпеля; д – ніпель з різьбою; е – заготовка муфти; ж – розточування муфти; з – профіль різьби муфти; і – муфта з різьбою; к – з'єднання

Рисунок 1 – Послідовність побудови геометричної моделі в Cadquery

зання, створено циліндричну виточку на ніпелі (рис. 1б) та зарізьбову канавку ніпеля (рис. 1в). Для створення профілю різьби ніпеля потрібно спершу створити профіль різьбового різця, що призначений для формування цієї різьби (рис. 1г). "Точіння" різьби реалізується шляхом відносного переміщення профілю інструмента та формування ним гвинтових поверхонь (канавок) у заготовці (рис. 1д).

Циліндрична заготовка муфти створюється у вигляді труби (рис. 1е). Розточування муфти виконують шляхом вирізання (рис. 1ж). Профіль внутрішнього різьбонарізного різця (рис. 1з) використовується для "точіння" різьби муфти шляхом переміщення і вирізання (рис. 1і). Потім ніпель і муфта об'єднуються в одну деталь (рис. 1к).

Для прикладу нижче наведемо частину коду, що створює зарізьбову канавку ніпеля (рис. 1б).

```
result2 = (cq.Workplane("XY")
    .hLine(p.dn-p.dln)
    .vLine(p.l2n)
    .hLine(-(p.dn-p.dln))
    .close()
    .extrude(2)
    .edges("|Z")
    .edges("<X")
    .fillet(p.r3n)
    .translate((p.dln,p.l1n-p.l2n,0))
)
result=result.cut(result2)
```

Тут result2 є формою, якою в заготовці result вирізають канавку. Ескіз будують в площині "XY". Побудова горизонтальних та вертикальних ліній здійснюється за допомогою функцій hLine та vLine, а p – це Python-модуль, що містить дійсні значення параметрів з'єднання (таблиця 1). За допомогою "|Z" та "<X" знаходяться усі ребра, які паралельні до осі Z та розташовані ліворуч вздовж осі X. На них створюється скруглення (fillet). Функція translate використовується для переміщення, а cut – для вирізання.

Модуль gost13877_19.py містить параметри муфтового різьбового з'єднання ШН19 за ГОСТ 13877-96 (табл. 1). Значення параметрів є кортежами з трьох елементів: (номінальний розмір, нижнє відхилення, верхнє відхилення).

Модуль gost13877.py розраховує фактичні значення параметрів, необхідних для генерації конкретної моделі. Алгоритм орієнтований на найбільш критичний стан з'єднання, за якого розміри різьби ніпеля є мінімально допустимими, а муфти – максимально допустимими. Таке поєднання параметрів зумовлює утворення максимальних зазорів, що призводить до найвищої концентрації напружень. Для цього модуль імпортує клас Dim з модуля tools та імпортує модуль gost13877_19 з параметрами. Клас Dim описує поняття розміру з такими його атрибутами як номінальний розмір, нижнє відхилення, верхнє відхилення, дійсне

Таблиця 1 – Параметри муфтового різьбового з'єднання ШН19 за ГОСТ 13877-96 в модулі gost13877_19.py

Назва	Змінна	Номінальний розмір, мм	Нижнє відхилення, мм	Верхнє відхилення, мм
Діаметр штанги	d0	19,1	-0,41	0,2
Крок різьби	p_n=p_m	2,54	-	-
Параметри ніпеля:				
Зовнішній діаметр різьби	d_n	27	-0,48	-0,376
Середній діаметр різьби	d2_n	25,35	-0,204	-0,047
Внутрішній діаметр різьби	d1_n	24,25	0	-0,415
Радіус западин різьби	r_n	0,28	0	0,08
Діаметр бурта	dn	38,1	-0,25	0,13
Діаметр зарізьбової канавки	d1n	23,24	-0,13	0,13
Довжина ніпеля	l1n	36,5	0	1,6
Довжина зарізьбової канавки	l2n	15	0,2	1
Довжина ніпеля без фаски на різьбі	l3n	32	0	1,5
Довжина ніпеля з буртом	l4n	48	-1	1,5
Радіуси зарізьбової канавки	r3n	3	0	0,8
Параметри муфти:				
Зовнішній діаметр різьби	d_m	27	0	0,27
Середній діаметр різьби	d2_m	25,35	0	0,202
Внутрішній діаметр різьби	d1_m	24,25	0	0,54
Зовнішній діаметр	dm	41,3	-0,25	0,13
Внутрішній діаметр опорної поверхні	d1m	27,43	0	0,25
Довжина муфти	lm	102	-1	1

значення, мінімальний розмір, максимальний розмір. Оскільки модель є осесиметричною, то для зручності діаметральні розміри перетворюються в радіальні:

```
d_n = r.d_n.min() / 2 # зовнішній радіус різьби
d2_n = r.d2_n.min() / 2 # середній радіус різьби
```

Також модуль може містити допоміжні параметри, які полегшують побудову моделі. Наприклад, характерні точки для ідентифікації ребер чи вершин моделі.

Виклик функції `run(p, name)` модуля `run_sq2.py` будує геометричну осесиметричну модель з'єднання з параметрами `p` та іменем `name` та конвертує її у формат DXF. Файл `<name>.dxf` зберігається на диску у робочому каталозі. Модуль може бути імпортований або запущений. Приклад запуску функції `run`, що будує модель стандартного з'єднання, але з параметром `r3n=1.6` мм:

```
import gost13877 as p
p.r3n=1.6
run(p, 'model')
```

Для візуалізації результатів (рис. 2) під час запуску з середовища CQ-editor необхідно використати функцію `show_object`.

Модуль `run_руссх3.py` [24] містить функцію `run(p, name)`, що будує та обчислює скінченно-елементну модель за допомогою Русалculix з використанням параметрів `p`. Формується вхідний файл для Calculix з назвою `<name>.inp`. Усі файли моделі під іменем `<name>` зберігаються в поточному каталозі. Нижче описано роботу функції `run`.

Спочатку `run` створює модель з іменем `name`, установлює одиниці вимірювання, виконує імпорт файлу `<name>.dxf` з геометрією (рис. 3а) та створює деталь.

```
model = pyc.FeaModel(name) #
модель
model.set_units('mm') # одиниці
вимірювання
importer =
pyc.CadImporter(model,
fname=name+".dxf", swapxy=False)
parts = importer.load()
part=parts[0]
```

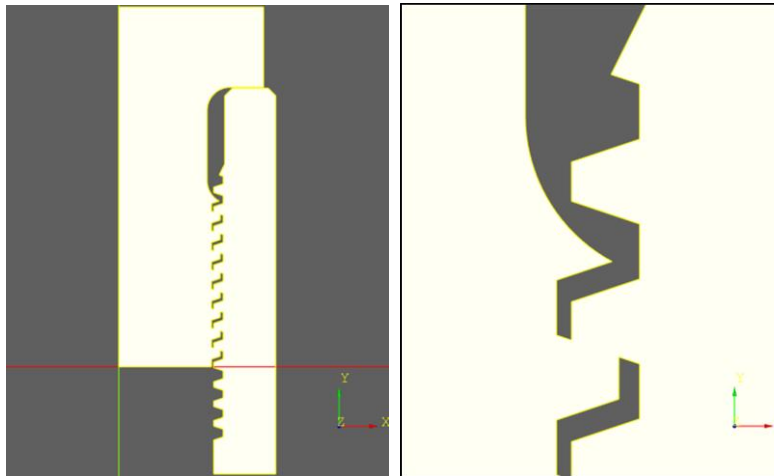



Рисунок 2 – Візуалізація геометричної моделі в CQ-editor

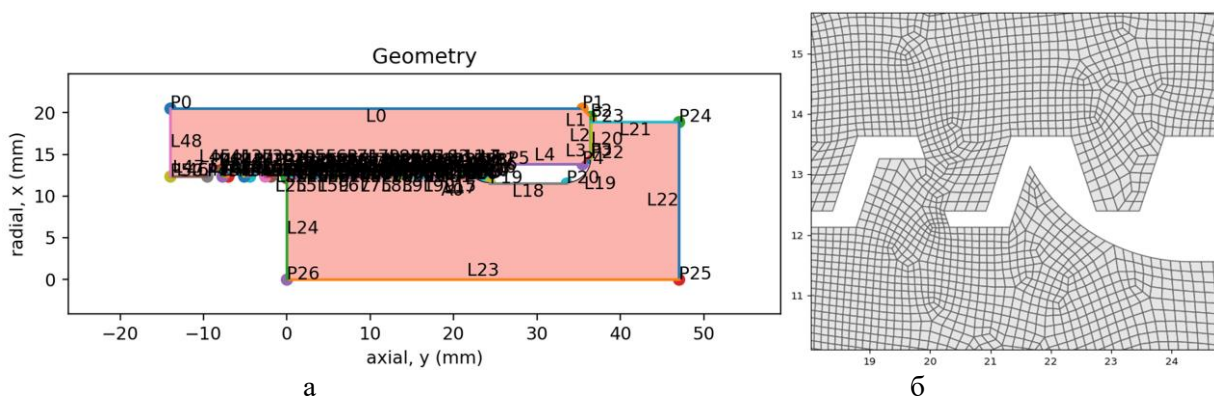


Рисунок 3 – Результат імпорту геометрії з DXF-файлу (а) та сітка елементів в зоні першого витка (б)

У даному випадку існує тільки одна деталь (part), бо ніпель і муфта є одним тілом з невеликим зазором на упорному бурті. Це зроблено для спрощення задачі та уникнення необхідності задавати умови контактної взаємодії між тілами. Також зусилля згвинчування з'єднання не моделюється. Це той аварійний випадок, коли різьбове з'єднання втратило зусилля згвинчування (самостійно відгвинтілось).

Далі вказується матеріал – сталь з модулем пружності $2,1 \cdot 10^{11}$ Па та коефіцієнтом Пуассона 0,3. Пластичні деформації в реальних з'єднаннях спостерігаються майже завжди [1], а Calculix дозволяє створювати нелінійні моделі матеріалів з пластичністю [15]. Проте, з метою пришвидшення обчислень, це дослідження не враховує властивості пластичності матеріалу.

```
mat = pyc.Material('steel')
mat.set_mech_props(7800, 2.1e11,
0.3) # властивості матеріалу
model.set_mat1(mat, part)
```

Тип елементів сітки — осесиметричний, форма елементів — чотирикутна (рис. 3б). Сітка створюється автоматично засобами GMSH. Для отримання точніших результатів можливо зменшити крок розбиття елементів

```
model.set_etype('axisym', part)
# осесиметричні елементи
model.set_eshape('quad', 1) #
форма елементів - tri, quad
model.mesh(0.1, 'gmsht') # створити сітку
```

Далі формулюються граничні умови і навантаження. Деталь позбавлена переміщень і поворотів на лівому торці муфти (відрізок L48), а правий торець ніпеля (відрізок L22) переміщується вздовж осі Y на 0,01 мм (рис. 3а). Оскільки програма не знає імен цих відрізків, то для пошуку лінії за координатами її середньої точки застосовано функцію findLine, яка повертає список імен знайдених ліній. Шукаються лінії Lfix та Lpress:

```

Lfix=list(findLine(lines=model.
lines, midpt=((p.dl_m+p.dm)/2, -
(p.lm-p.l1n))))[0].get_name()
Lpress=list(findLine(lines=model.
lines, midpt=(p.dn/2, p.l4n)))[0].
get_name()
model.set_constr('fix', [Lfix],
'x')
model.set_constr('fix', [Lpress],
'y')
model.set_constr("disp", [Lpress],
'y', 0.01)

```

Модифікована авторами функція `prob.solve(fn=edit_inp)` з пакету `Rucalcuix` дозволяє через функцію `edit_inp(filename)` формувати контактні задачі перед симуляцією. Оскільки в даному дослідженні контакту поверхонь немає, то ця функція є пустою. Виклик функції `prob.solve()` потрібно розташувати в блоці `try-except`, щоб помилки виконання не завершували аварійно роботу програми.

```

prob = ruc.Problem(model,
'struct')
try:
    prob.solve(fn=edit_inp)
except:
    print('solver error!')

```

Після завершення обчислень відбувається зчитування результатів у момент часу 1 (останній підкрок статичного аналізу): `prob.rfile.set_time(1.0)`. Користувач також може переглянути результати через постпроцесор `cgx.exe`, якщо їй передати файл результатів з розширенням `FRD`.

В модулі передбачено команди для візуалізації геометрії, сітки та результатів. Ці команди потрібно відключити, якщо модуль використовується для задачі оптимізації в мультиагентному середовищі. Функція `run` повертає максимальне еквівалентне напруження в конструкції (`res=prob.rfile.get_nmax('Seqv')`). Якщо зчитування результатів призводить до помилки, то функція повертає 0. Приклад запуску:

```

import gost13877 as p
p.r3n=3.8 # установити значення
параметра
print(run(p))

```

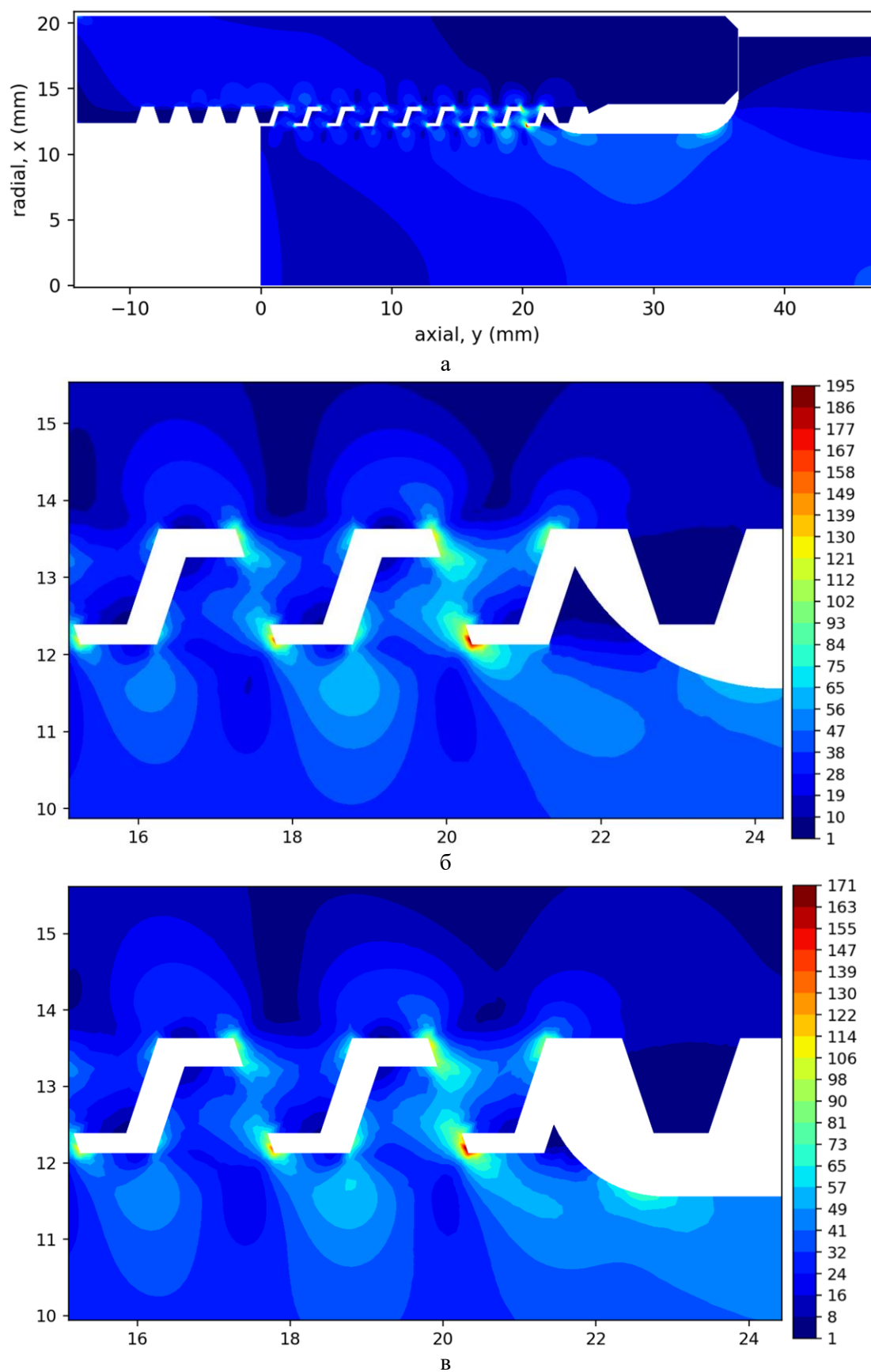
Після обчислення функція повертає максимальне еквівалентне напруження за критерієм Мізеса для стандартного з'єднання – 195 МПа. Воно локалізується в зоні западини першого витка ніпеля (рис. 4а,б), що узгоджується з відомими дослідженнями [1, 2, 3, 9-11]. Зменшення радіуса зарізьбової канавки до 1,6 мм призводить до зменшення напружень у цій зоні до 171 МПа. Це можна пояснити збільшенням податливості зарізьбової канавки. В самій зарізьбовій канавці напруження дещо зростають (рис. 4в).

Модуль `opt2_2.py` [24] є головним, використовує усі описані вище компоненти (рис. 5) та містить класи акторів для оптимізації конструкції різьбового з'єднання за критерієм мінімальних еквівалентних напружень. Для паралельних обчислень використано модель акторів, що реалізована за допомогою пакету `Ray`. Класи містять декоратори `@ray.remote`, які їх перетворюють у повноцінні класи акторів з можливістю паралельної роботи. Оптимізаційні обчислення базуються на мультиагентному підході. Існує середовище $Y=f(X)$, в якому різні агенти оперують різними значеннями параметрів X і з використанням різних правил поведінки в цьому середовищі (функція `rule`) намагаються знайти мінімум функції $\min f(X)$.

Клас `FEA` дозволяє створювати акторів-агентів, правилом (`rule`) яких є обчислення максимального еквівалентного напруження в з'єднанні за значеннями його геометричних параметрів x . В даному випадку це радіус зарізьбової канавки (`r3n`) та зовнішній радіус муфти (`dm`). Для роботи функція `rule` використовує модулі `gost13877` (параметри), `run_cq2` (геометрія), `run_rucscx3` (FEA-модель). Оскільки агенти працюють паралельно, то важливо, щоб імена файлів моделей, які створюють актори, були різними. Тому імена файлів містять унікальний ідентифікатор об'єкта `id(self)`.

Клас `Opt` є базовим абстрактним класом оптимізаційних агентів, правило яких намагається знайти мінімум функції. Усі агенти володіють атрибутом `bounds`, що містить границі значень параметрів x , та атрибутом `actorFEA`, що є FEA-актором. Також є атрибуту-функції для отримання значення `y` (`gety`) та обчислення значення `y` з використанням актора `actorFEA` (`calcy`).

Клас `Opt1` успадковує клас `Opt` та призначений для пошуку мінімуму функції з використанням сіткового методу ("brute force").



$a, б - r_{3n}=3,8 \text{ мм}; в - r_{3n}=1,6 \text{ мм}$

Рисунок 4 – Еквівалентні напруження за Мізесом (МПа) в з'єднанні (а) та в зоні першого витка (б, в)

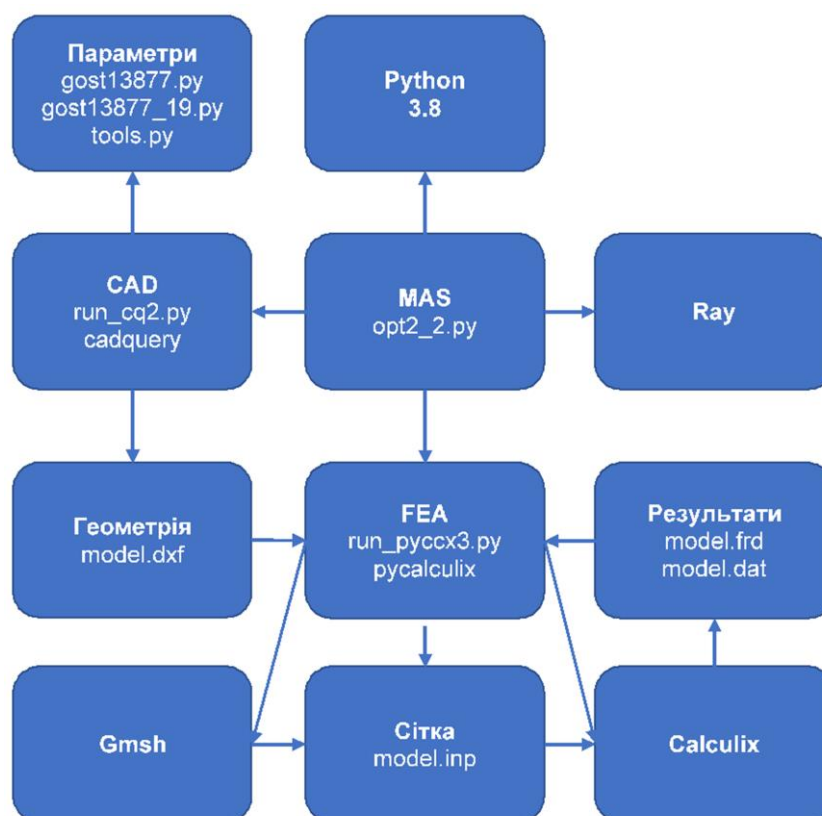


Рисунок 5 – Схема розробленої системи

Сітка значень X вказується перед запуском алгоритму. Функція `rule` шукає в середовищі (X,Y) точки (x,y) з невизначеними значеннями $(y=np.nan)$ та обчислює y за відомими x . Обчислену точку (x,y) функція повертає в середовище.

Клас `Opt2` також успадковує клас `Opt` та призначений для пошуку мінімуму функції з використанням стохастичного пошуку. Його функція `rule` в середовищі (X,Y) створює випадкову точку (x,y) і обчислює в ній значення y . Існують різні стратегії випадкового пошуку. Найпростіша з них передбачає генерування випадкового числа в заданих межах на основі рівномірного розподілу. Більш складні підходи базуються на пошуку в околі поточного глобального мінімуму (*argmin*, *min*) застосуванням рівномірного або нормального розподілів. Щоб уникнути потрапляння в пастку локального мінімуму доцільно проводити пошук також в околі інших точок із низькими значеннями цільової функції. Випадковий вибір цих точок здійснюється випадковим чином відповідно до заданого розподілу ймовірностей, що забезпечує необхідне різноманіття пошукових стратегій.

Інші похідні класи `Opt` можуть реалізовувати, для прикладу, градієнтні методи, методи

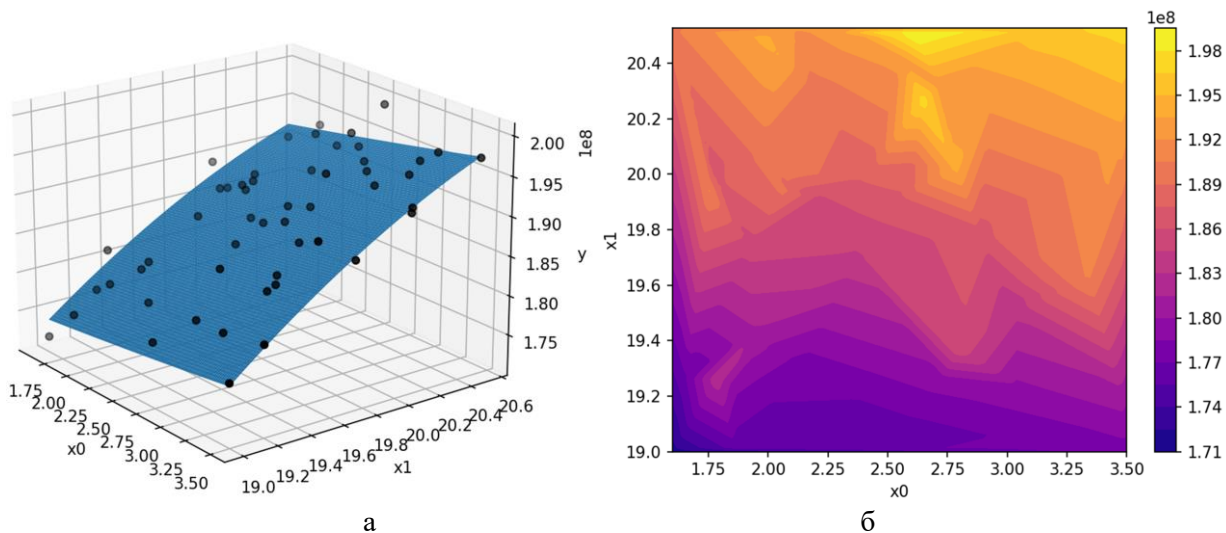
інтерполяції або апроксимації або шукати мінімум в місцях з найменшою густиною точок. Такий мультиагентний підхід дозволяє знаходити найбільш вдалі поєднання різнотипних стратегій пошуку мінімуму.

Клас `Environment` описує середовище (X,Y) , в якому агенти намагаються знайти мінімум значень напружень. Клас володіє атрибутами: X , Y – масиви для збереження аргументів та значень функції, `getXY` – функція для читання значень X , Y , `setXY` – функція для запису усіх значень X , Y , `addxy` – функція для додавання однієї точки (x,y) в середовище, `sety` – функція для запису значення y , якщо воно було рівне `np.nan` (не число).

Для мультиагентної паралельної оптимізації спочатку вказуються границі значень параметрів, де перша пара вказує границі для $r3n$, а друга – для dm (мм):

```
bounds=(1.6, 3.5), (19.0, 20.525)
```

Створюється початкова сітка для агентів `Opt1`. Спочатку значення y є невідомими, тому їм присвоюється `np.nan`. Після цього створюється один актор класу `Environment` та актори-агенти класів `Opt1` (чотири актори)



x_0 – радіус зарізьбової канавки (мм), x_1 – зовнішній радіус муфти (мм),
 y – еквівалентні напруження (Па)

Рисунок 6 – Залежності $Y=f(X)$, отримані шляхом апроксимації (а) та інтерполяції (б) результатів симуляції

та Opt2 (два актори). Кількість може бути іншою, що залежить від обраної стратегії оптимізації та кількості ядер центрального процесора чи кластера.

```
e=Environment.remote(X, Y)
O = [Opt1.remote(bounds) for i
in [0,1,2,3]]
O+= [Opt2.remote(bounds) for i
in [0,1]]
```

Далі розпочинається оптимізаційний цикл, що містить 40 ітерацій. На кожній ітерації паралельно запускається функція rule кожного агента. Після завершення циклу значення X,Y середовища зберігаються у файлах.

```
for i in range(40):
    ray.get([o.rule.remote()
for o in O])
X,Y=ray.get(e.getXY.remote())
```

Результати (рис. 6а) показують, що мінімум знаходиться в точці (1,6; 19,0). Це можна пояснити тим, що зменшення товщини муфти, наприклад внаслідок зношування її поверхні під час тертя об насосно-компресорну трубу, робить її податливою і це зменшує навантаження на витки ніпеля. Проте напруження в останньому витку різьби муфти зростають. Зменшення ж радіуса зарізьбової канавки робить її більш податливою і теж зменшує напруження у перших витках ніпеля.

На рис. 6,а показана апроксимація усіх даних поліноміальною залежністю

$$y = ax_0^2 + bx_1^2 + cx_0x_1 + dx_0 + ex_1 + f, \quad (1)$$

де $a=1.27950294e+05$; $b=-2.99343478e+06$;
 $c=6.45753230e+05$; $d=-1.11507385e+07$;
 $e=1.29095147e+08$; $f=-1.20024463e+09$.

Коефіцієнт детермінації R^2 має достатньо високе значення 0,915, що дозволяє використовувати апроксимацію для обґрунтування проектних рішень.

Лінійна інтерполяція даних (рис. 6,б) може використовуватись в алгоритмі оптимізації для обґрунтування вибору перспективних точок.

Висновки

Запропоновані принципи створення та використання вільної мультиагентної CAD/FEA-системи, що призначена для автоматизації багатовимірної параметричної оптимізації конструкції різьбового з'єднання насосних штанг.

1. Розроблений на основі Cadquery модуль для автоматизації побудови геометричної моделі різьбового з'єднання дозволяє легко змінювати параметри конструкції і модифікувати модель. Розроблені моделі можуть поширюватись і розвиватись з використанням систем керування версіями програмного забезпечення.

2. Удосконалений на основі Ryscalulix модуль для побудови осесиметричної скінченно-елементної моделі дозволяє легко імпортувати геометричну модель, створити сітку, виконати симуляцію і прочитати результати в автоматично.

чному режимі. Цей модуль призначений тільки для FEA без моделювання контакту поверхонь, проте є можливість змоделювати контакт в майбутньому.

3. З використанням пакету Ray та моделі акторів удосконалено мультиагентну систему для параметричної оптимізації конструкції різьбового з'єднання за критерієм мінімальних напружень. Система містить агенти з протилежними правилами (сітковий пошук та стохастичний пошук), що дозволяє ефективно шукати глобальні мінімуми в багатовимірних просторах. Існує також можливість простого створення агентів з іншими правилами і удосконалення стратегії оптимізації.

4. З використанням цієї системи досліджено напружено-деформований стан різьбового з'єднання ШН19 в аварійних умовах відсутнос-

ті попереднього затягування та виявлено вплив на напруження радіуса зарізьбової канавки та зовнішнього діаметра муфти. Зменшення цих параметрів призводить до зменшення еквівалентних напружень у найбільш навантаженій зоні першого витка.

5. Код розробленої системи автоматизованого проектування опубліковано на GitHub під вільною ліцензією GPL-3.0 [24], що дозволяє їй ефективно розвиватися.

Подяки

Відсутні.

Конфлікт інтересів

Відсутній.

Список використаних джерел

1. Sucker rod failure analysis. CHAMPIONX Artificial lift. Tulsa, USA : Norris Rods. URL: https://www.championx.com/contents/NOR_Sucker%20Rod%20Failure%20Analysis_BR_0322.pdf
2. Копей В. Б., Копей Б. В. Проектування та міцність колон і з'єднань насосних штанг з полімерних композитів : монографія. Серія «Нафтогазове обладнання» : у 12 т. За заг. ред. Б. В. Копея. Том 12. Івано-Франківськ : ІФНТУНГ, 2021. 314 с.
3. Fukuoka T. The Mechanics of Threaded Fasteners and Bolted Joints for Engineering and Design. Amsterdam, The Netherlands : Elsevier, 2022. 382 p.
4. Мойсшин В. М., Сіренко В. В. Обсадні труби та їх з'єднання: аналіз сортаменту, оцінка напружено-деформованого стану. *Науковий вісник ІФНТУНГ*. 2010. № 1(23). С. 57–66. URL: <https://nv.nung.edu.ua/index.php/nv/article/view/56>
5. Kopey V. V., Maksymuk A. V., Shcherbyna N. N. Analysis of contact stresses in structural joints of composite shell with steel banding. *Mechanics of Composite Materials*. 2000. № 36(1). P. 67–74. DOI: <https://doi.org/10.1007/BF02681779>
6. Shatskyi I., Ropyak L., Velychkovych A. Model of contact interaction in threaded joint equipped with spring-loaded collet. *Engineering Solid Mechanics*. 2020. № 8. P. 301–312. DOI: <https://doi.org/10.5267/j.esm.2020.4.002>
7. Medvid I., Onysko O., Pituley L., Odosii Z., Kornuta O. The Influence of the Geometry of High-Performance Cutters on the Profile Accuracy of Large-Pitch Tapered Threads: Theoretical Study. Advances in Design, Simulation and Manufacturing VII. DSMIE 2024. *Lecture Notes in Mechanical Engineering*. Cham : Springer, 2024. DOI: https://doi.org/10.1007/978-3-031-61797-3_25
8. Balytskyi O. I., Kolesnikov V. O., Elias J. Study of the wear resistance of high-nitrogen steels under dry sliding friction. *Materials Science*. 2013. № 48. P. 642–646. DOI: <https://doi.org/10.1007/s11003-013-9549-7>
9. Hoffman E. L. Finite Element Analysis of Sucker Rod Couplings with Guidelines for Improving Fatigue Life : Sandia report. Sandia National Laboratories, 1997. 66 p. DOI: <https://doi.org/10.2172/537261>
10. Михайлюк В. В. Розроблення різьбового з'єднання насосних штанг з рівномірним розподілом зусиль між витками різьби. *Науковий вісник ІФНТУНГ*. 2010. № 4(26). С. 61–65. URL: <https://nv.nung.edu.ua/index.php/nv/article/view/460>
11. Михайлюк В. В., Юрич А. Р., Дейнега Р. О., Кравчук Р. С., Пінчак Р. А., Харламов Б. В. Вдосконалення та дослідження різьбового з'єднання насосних штанг. *Науковий вісник ІФНТУНГ*. 2018. № 1(44). С. 81–88. DOI: [https://doi.org/10.31471/1993-9965-2018-1\(44\)-81-88](https://doi.org/10.31471/1993-9965-2018-1(44)-81-88)
12. Zhu X., Zhi Z. Design of an ultra-high torque double shoulder drill-pipe tool joint for extended reach wells. *Natural Gas Industry B*. 2017. Vol. 4, Iss. 5. P. 374–381. DOI: <https://doi.org/10.1016/j.ngib.2017.10.002>

13. Копей В. Б., Онисько О. Р., Жигуд Ю. Ю. Обґрунтування застосування двоопорних різьбових з'єднань пустотілих насосних штанг. *Науковий вісник ІФНТУНГ*. 2019. № 1(46). С. 7–15. DOI: [https://doi.org/10.31471/1993-9965-2019-1\(46\)-7-15](https://doi.org/10.31471/1993-9965-2019-1(46)-7-15)
14. Increasing the fatigue strength of threaded joints of oil and gas equipment by plastic deformation of the thread under high load before make-up / V. B. Kopei, O. R. Onysko, V. G. Panchuk, Z. M. Odosii, Y. M. Kusi. *Journal of Physics: Conference Series*. 2023. Vol. 2540. P. 012033. DOI: <https://doi.org/10.1088/1742-6596/2540/1/012033>
15. Dhondt G. The Finite Element Method for Three-Dimensional Thermomechanical Applications. Hoboken, NJ, USA : Wiley, 2004. DOI: <https://doi.org/10.1002/0470021217>
16. Geuzaine C., Remacle J. F. Gmsh: A Three-Dimensional Finite Element Mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*. 2009. Vol. 79, Iss. 11. P. 1309–1331. DOI: <https://doi.org/10.1002/nme.2579>
17. Black J. Pycalculix. URL: <https://github.com/spacether/pycalculix> (дата звернення: 25.12.2025).
18. Open CASCADE Technology. Collaborative development portal. URL: <https://dev.opencascade.org> (дата звернення: 25.12.2025).
19. CadQuery contributors. CadQuery (v2.6.1). Zenodo, 2025. DOI: <https://doi.org/10.5281/zenodo.14590990>
20. Metaheuristic. Wikipedia. URL: <https://en.wikipedia.org/wiki/Metaheuristic> (дата звернення: 25.12.2025).
21. Wooldridge M., Jennings N. R. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*. 1995. Vol. 10, Iss. 2. P. 115–152. DOI: <https://doi.org/10.1017/S0269888900008122>
22. Designing a Multi-Agent PLM System for Threaded Connections Using the Principle of Isomorphism of Regularities of Complex Systems / V. Kopei, O. Onysko, C. Barz, P. Dašić, V. Panchuk. *Machines*. 2023. Vol. 11(2). P. 263. DOI: <https://doi.org/10.3390/machines11020263>
23. Ray: A Distributed Framework for Emerging AI Applications / P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan et al. *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, Carlsbad, CA, USA, 2018. P. 561–577. URL: <https://www.usenix.org/conference/osdi18/presentation/moritz>
24. Kopei V., Protsiv O. ThreadMAS. URL: <https://github.com/vkopey/ThreadMAS> (дата звернення: 22.12.2025).

References

1. ChampionX Artificial Lift. (n.d.). *Sucker rod failure analysis*. Norris Rods. https://www.championx.com/contents/NOR_Sucker%20Rod%20Failure%20Analysis_BR_0322.pdf
2. Kopei, V. B., & Kopei, B. V. (2021). *Design and strength of sucker rod strings and connections made of polymer composites: Monograph* (B. V. Kopei, Ed.; Vol. 12). IFNTUNG. [in Ukrainian]
3. Fukuoka, T. (2022). *The mechanics of threaded fasteners and bolted joints for engineering and design*. Elsevier.
4. Moysyshyn, V. M., & Sirenko, V. V. (2010). Casing pipes and their connections: assortment analysis, assessment of stress-strain state. *Scientific Bulletin of IFNTUNG*, (1), 57–66. <https://nv.nung.edu.ua/index.php/nv/article/view/56> [in Ukrainian]
5. Kopey, B. V., Maksymuk, A. V., & Shcherbyna, N. N. (2000). Analysis of contact stresses in structural joints of composite shell with steel banding. *Mechanics of Composite Materials*, 36(1), 67–74. <https://doi.org/10.1007/BF02681779>
6. Shatskyi, I., Ropyak, L., & Velychkovych, A. (2020). Model of contact interaction in threaded joint equipped with spring-loaded collet. *Engineering Solid Mechanics*, (8), 301–312. <https://doi.org/10.5267/j.esm.2020.4.002>
7. Medvid, I., Onysko, O., Pituley, L., Odosii, Z., & Kornuta, O. (2024). The influence of the geometry of high-performance cutters on the profile accuracy of large-pitch tapered threads: Theoretical study. In *Advances in Design, Simulation and Manufacturing VII. DSMIE 2024* (Lecture Notes in Mechanical Engineering). Springer. https://doi.org/10.1007/978-3-031-61797-3_25
8. Balytskyi, O. I., Kolesnikov, V. O., & Elias, J. (2013). Study of the wear resistance of high-nitrogen steels under dry sliding friction. *Materials Science*, (48), 642–646. <https://doi.org/10.1007/s11003-013-9549-7>

9. Hoffman, E. L. (1997). *Finite element analysis of sucker rod couplings with guidelines for improving fatigue life* (Sandia Report). Sandia National Laboratories. <https://doi.org/10.2172/537261>
10. Mykhailiuk, V. V. (2010). Development of sucker rod threaded connection with uniform distribution of forces between the threads. *Scientific Bulletin of IFNTUNG*, (4), 61–65. <https://nv.nung.edu.ua/index.php/nv/article/view/460> [in Ukrainian]
11. Mykhailiuk, V. V., Yurvch, A. R., Deineha, R. O., Kravchuk, R. S., Pinchak, R. A., & Kharlamov, B. V. (2018). Improvement and research of sucker rod threaded connection. *Scientific Bulletin of IFNTUNG*, (1), 81–88. [https://doi.org/10.31471/1993-9965-2018-1\(44\)-81-88](https://doi.org/10.31471/1993-9965-2018-1(44)-81-88) [in Ukrainian]
12. Zhu, X., & Zhi, Z. (2017). Design of an ultra-high torque double shoulder drill-pipe tool joint for extended reach wells. *Natural Gas Industry B*, 4(5), 374–381. <https://doi.org/10.1016/j.ngib.2017.10.002>
13. Kopei, V. B., Onysko, O. R., & Zhyhuts, Y. Y. (2019). Substantiation of the use of double-shoulder threaded connections of hollow sucker rods. *Scientific Bulletin of IFNTUNG*, (1), 7–15. [https://doi.org/10.31471/1993-9965-2019-1\(46\)-7-15](https://doi.org/10.31471/1993-9965-2019-1(46)-7-15) [in Ukrainian]
14. Kopei, V. B., Onysko, O. R., Panchuk, V. G., Odosii, Z. M., & Kusyi, Y. M. (2023). Increasing the fatigue strength of threaded joints of oil and gas equipment by plastic deformation of the thread under high load before make-up. *Journal of Physics: Conference Series*, 2540, 012033. <https://doi.org/10.1088/1742-6596/2540/1/012033>
15. Dhondt, G. (2004). *The finite element method for three-dimensional thermomechanical applications*. Wiley. <https://doi.org/10.1002/0470021217>
16. Geuzaine, C., & Remacle, J. F. (2009). Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331. <https://doi.org/10.1002/nme.2579>
17. Black, J. (2025). *Pycalculix*. GitHub. <https://github.com/spacether/pycalculix>
18. Open CASCADE Technology. (2025). *Collaborative development portal*. <https://dev.opencascade.org>
19. CadQuery contributors. (2025). *CadQuery* (v2.6.1) [Software]. Zenodo. <https://doi.org/10.5281/zenodo.14590990>
20. Wikipedia. (2025). *Metaheuristic*. <https://en.wikipedia.org/wiki/Metaheuristic>
21. Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152. <https://doi.org/10.1017/S0269888900008122>
22. Kopei, V., Onysko, O., Barz, C., Dašić, P., & Panchuk, V. (2023). Designing a multi-agent PLM system for threaded connections using the principle of isomorphism of regularities of complex systems. *Machines*, 11(2), 263. <https://doi.org/10.3390/machines11020263>
23. Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., et al. (2018). Ray: A distributed framework for emerging AI applications. *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, 561–577. <https://www.usenix.org/conference/osdi18/presentation/moritz>
24. Kopei, V., & Protsiv, O. (2025). *ThreadMAS*. GitHub. <https://github.com/vkopey/ThreadMAS>

MULTI-AGENT SYSTEM FOR THE AUTOMATED DESIGN OF SUCKER ROD THREADED CONNECTIONS

Kopey V. B.

Doctor of Technical Sciences, Professor
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska Str., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0000-0003-0008-8260>
e-mail: vkopecy@gmail.com

Protsiv O. I.

Student
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska Str., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0009-0008-5204-5055>
e-mail: oleksandr.protsiv-pmkm241@nung.edu.ua

Proniuk I. V.

Postgraduate
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska Str., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0009-0005-1029-5337>
e-mail: pronyuk00@gmail.com

Nazarenko S. K.

Postgraduate
Ivano-Frankivsk National Technical University of Oil and Gas
76019, Karpatska Str., 15, Ivano-Frankivsk, Ukraine
<https://orcid.org/0009-0001-0787-8655>
e-mail: 1314296@ukr.net

Abstract. Threaded connections of sucker rods are used in oil production with rod pumping units and are designed to connect individual rods into a single string. During the operation of sucker rod threaded connections, numerous problems arise due to cyclic loads and high stress concentration in the thread region, leading to fatigue cracks, failures, and string breakage. The strength of these connections is influenced by many design factors, which can be investigated using the finite element method. This paper outlines the principles for developing and utilizing a multi-agent system for the automated design of sucker rod threaded connections. To create modules for automating the construction of geometric and axisymmetric finite element models of the threaded connection, open-source software CalculiX, Gmsh, and Python packages CadQuery and PyCalculiX were used. These packages significantly simplify the creation of parametric finite element models suitable for fully automated design optimization based on stress and strain criteria. Using these modules and an actor-based model for parallel computing (based on the Python package Ray), a multi-agent system was enhanced for the parametric optimization of the threaded connection design according to the criterion of minimum equivalent stress. The system incorporates agents with opposing optimization rules (implementing grid and stochastic search methods), enabling efficient search for minima in multi-dimensional spaces. There is also the possibility of easily creating agents with other rules and finding their effective combinations. Using this system, the stress-strain state of the ShN19 threaded connection was investigated under emergency conditions of missing preload, which occurs in case of connection self-loosening. The influence of the thread runout groove radius and the coupling outer diameter on equivalent stresses was also studied. The code of the developed automated design system is published on GitHub under the free GPL-3.0 license.

Keywords: threaded connection, automated design, multi-agent system, actor model, optimization, stress.