Доля П.Г.
Харківський національний університет імені В. Н. Каразіна
факультет математики і інформатики
кафедра теоретичної і прикладної інформатики
2019 р.

Основи моделювання в COMSOL Multiphysics

Глава 1. Початок роботи в COMSOL Multiphysics.

Зміст

1. Початок роботи в COMSOL Multiphysics. 2
1.1. Вступ. 2
1.2. Опис інтерфейсу на прикладі розв'язання рівняння Пуассона. 4
1.3. Функції і графіки. 17
1.4. Термінологія, позначення і скорочення. 36
1.5. Початкові відомості про моделювання в COMSOL Multiphysics. 42
1.5.1. Структура моделей COMSOL Multiphysics. 42
1.5.2. Керування деревом моделі 65
1.5.3. Налаштування графіки. 72
1.5.4. Параметри, змінні, зонди (probes). 93
1.5.5. Побудова геометрії. 113
1.6. Демонстраційні приклади. 123
1.6.1. Охолодження куба. 123
1.6.2. Коливання вільно опертої балки. 130
1.6.3. Осесиметричні коливання кругової мембрани 135
1.7. Математичні вирази 140
1.8. Довідкова система і документація 162
1.9. Задачі для самостійного розв'язання. 164

Числове моделювання фізичних процесів є важливим етапом проектування конструкцій і технологій. Оскільки багато фізичних законів формулюється за допомогою диференціальних рівнянь в частинних похідних (ДРЧП), то для їх дослідження потрібен універсальний інструмент, який, крім іншого, здатний поєднувати різні фізичні моделі. Таке об'єднання можна назвати «мультифізикою», що означає спільне використання декількох моделей.

COMSOL Multiphysics це потужне програмне середовище, яке створено для моделювання фізичних процесів і здатне розв'язувати широкий спектр наукових і інженерних задач з багатьох сфер людської діяльності. При розв’язанні ДРЧП в системі здебільшого використовується метод скінченних елементів (МСкЕ). Система має налаштовані інтерфейси (прикладні режими) для розв’язання задач з різноманітних сфер, таких як механіка, електромагнетизм, гідродинаміка, хімія, акустика, теплопровідність, дифузія, оптика, опір матеріалів, теорія пружності і багатьох інших. COMSOL Multiphysics дозволяє поєднувати несхожі фізичні моделі і моделі різних розмірностей, а також дозволяє додавати до моделей власні рівняння. З самого початку, не використовуючи визначені інтерфейси, можна записати задачу в формі системи рівнянь, яка може включати алгебраїчні рівняння (АР), звичайні диференціальні рівняння (ЗДР) та ДРЧП. Можливості системи COMSOL Multiphysics величезні, і викласти їх в межах одного посібника нереально. Тут ми дамо вступ в розв’язання класичних задач. При цьому ми намагатимемося познайомити вас з системою на прикладах, розв’язок яких можливо відомий читачеві. Це полегшить знайомство з програмою. Але головна потужність пакету полягає в його прикладних режимах. Опис прикладів використання деяких з них буде надано в подальших главах.

Читати посібник ми радимо послідовно. Головні можливості системи викладаються в процесі розв’язання задач. Практично в кожному прикладі розглядаються нові елементи моделювання (опції, параметри, способи і т.д.). При цьому нові етапи моделювання викладаються докладно, але їх повторне використання в наступних задачах описується стисло. Не розібравши попередні приклади, ви, можливо, деякі з етапів згодом не зможете відтворити.

Якщо ви ще не вирішили , чи потрібно вам вивчати COMSOL Multiphysics, то познайомтесь з прикладами в наступних главах посібника. Ілюстрації до їх розв’язків надають уявлення про можливості системи.

1. Початок роботи в COMSOL Multiphysics.

1.1. Вступ.

В даному посібнику буде описуватися робота системи COMSOL Multiphysics версії 5.1, яка вийшла в 2015 році. До 2005 року вона називалася FEMLAB (Finite Element Method Laboratory). Історія програми починається з пакету-додатку PDE Toolbox MatLab, який призначений для розв’язання диференціальних рівнянь в частинних похідних і який досі входить до складу основної поставки MatLab. Друга версія так само, як і перша, не могла

працювати без MatLab, але вже постачалася як самостійний продукт. Починаючи з третьої версії, FEMLAB міг працювати як окрема програма або використовуватися як пакет додаток. Головним режимом роботі 4-ї та 5-ї версій є робота з графічним інтерфейсом користувача, але можливість спільної роботи з MatLab збереглася. При цьому синтаксис запису математичних виразів успадкований від першої версії і дуже близький до синтаксису MatLab.

Перша версія програми (PDE Toolbox MatLab) могла розв'язувати системи ДРЧП відносно функцій двох просторових змінних та змінної часу. Наступні версії могли вже розв'язувати тривимірні задачі. Пакет використовує метод скінченних елементів (МСЕ). В англомовній літературі він називається Finite Element Method (FEM). В цьому методі розрахункова область розбивається на велику кількість маленьких ділянок, які називаються скінченними елементами. Потім будуються фінітні функції з носіями, обмеженими кількома такими елементами. Ці функції утворюють базис в просторі кускових функцій на скінченно-елементній сітці. Шукані величини апроксимуються лінійною комбінацією базисних функцій. Крайова (гранична) задача для ДРЧП замінюється задачею пошуку мінімуму функціонала (деякого інтеграла по області), і в МСЕ зводиться до визначення коефіцієнтів розкладання апроксимуючих функцій по базисним.

На сайті розробників www.comsol.com/models розміщена галерея моделей, де можна докладно познайомитися з можливостями програми.

COMSOL Multiphysics вміє розв'язувати одновимірні (1D), двовимірні (2D) і тривимірні (3D) задачі. При цьому задачі можуть бути стаціонарними (шукані функції не залежать від часу), або динамічними (є залежність від часу). Інколи доводиться розглядати нульвимірні (0D) задачі, тобто розв'язувати просторово незалежні рівняння. Це означає, що задача зводиться до пошуку фізичних величин в точках. При цьому вони можуть бути сталими (стаціонарна модель) або функціями часу (динамічна модель). Чим вище розмірність задачі, тим складніше і триваліше розв'язання. У вісесиметричному випадку розмірність задачі можна понизити на одиницю. Для цього система виділяє випадки одновимірних і двовимірних осесиметричних задач (1D і 2D Axisymmetric). Таким чином, можна обирати один з шести можливих варіантів розмірності або їх поєднання.

Дуже важливою частиною програми є прикладні режими, які містять закінчені математичні постановки задач з різноманітних сфер фізики і техніки. Електронна версія документації по цим режимам встановлюється разом з COMSOL в двох форматах – PDF та HTML. Зокрема у встановлювальній папці системи в каталозі `..\doc\pdf\` містяться файли з документацією, які включають опис математичних постановок задач, процес моделювання та отримання результатів.

Головним режимом роботи програми є робота з графічним інтерфейсом користувача (режим COMSOL Desktop). Але існують і інші режими, наприклад, робота в командному вікні (режим COMSOL Batch) без використання графічного інтерфейсу. В цьому режимі заощаджується пам'ять за рахунок того, що вона не використовується під графічний інтерфейс. Існує

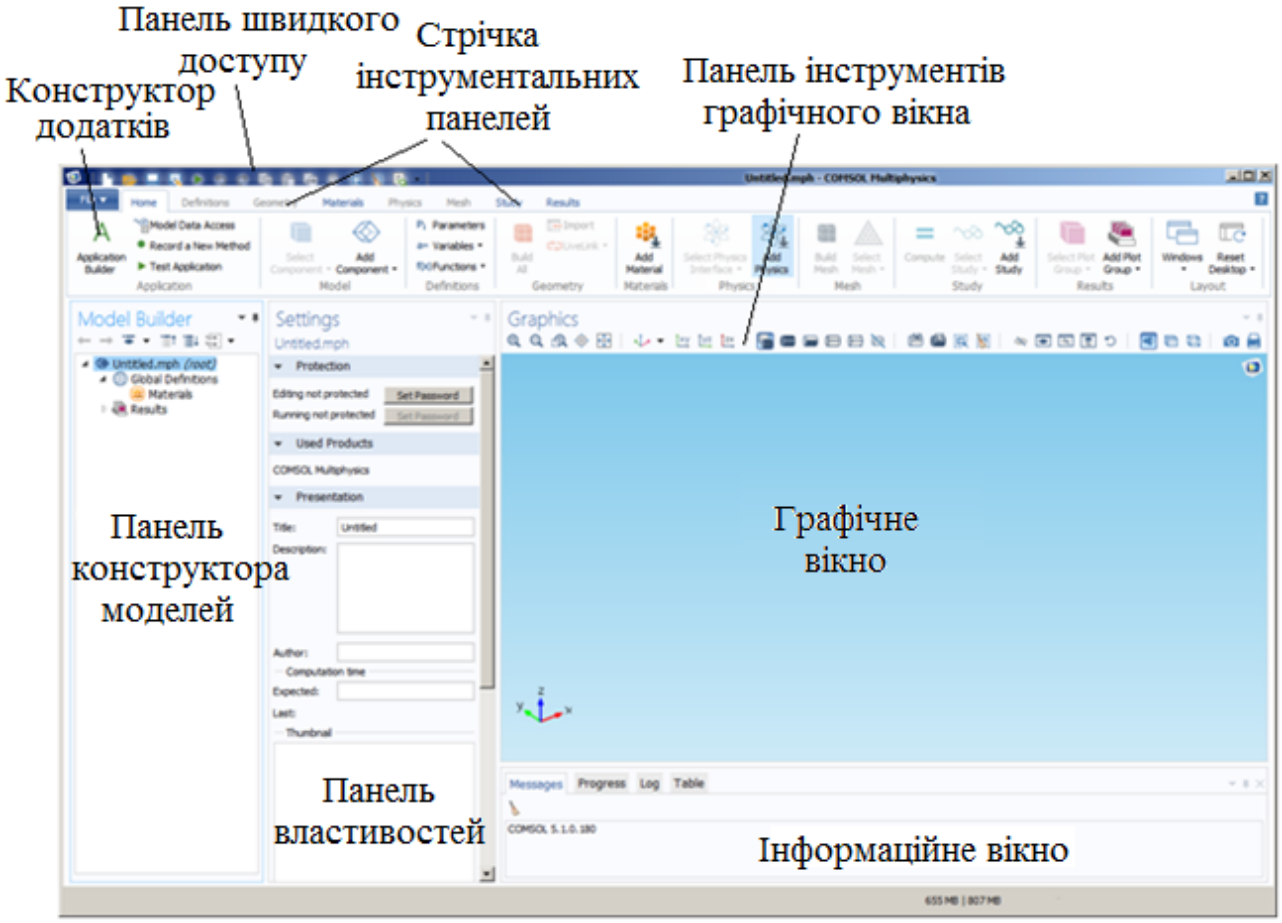
режим роботи з доступом до COMSOL Multiphysics з MatLab або, навпаки, з доступом з COMSOL Multiphysics до MatLab. Після встановлення системи, якщо до цього був встановлений MatLab, у вас на робочому столі з'явиться два ярлика. Перший «COMSOL Multiphysics 5.1» запускає систему з графічним інтерфейсом користувача. Другий ярлик, з назвою «COMSOL 5.1 with MATLAB», запускає MatLab з можливістю доступу до моделей COMSOL. Програма має інструменти для взаємодії з найбільш поширеними системами САПР, такими як Pro/ENGINEER (Creo Parametric), SOLIDWORKS, Autodesk Inventor, Autodesk Revit, Solid Edge. Існує можливість керувати моделями COMSOL з Excel.

В документації на прикладі мови програмування Java докладно описана можливість доступу до COMSOL API. В п'ятій версії програми з'явилося ще одне нововведення – середовище розробки застосунків (Application Builder), яке дозволяє конвертувати існуючу модель COMSOL в додаток.

Для багатоядерних та багатопроцесорних комп'ютерів існує можливість використовувати всі процесори і всі ядра.

1.2. Опис інтерфейсу на прикладі розв'язання рівняння Пуассона.

Після запуску програми ви побачите наступне вікно (ми припускаємо, що користувач не міняв налаштувань системи після інсталяції).



Панель швидкого доступу містить кнопки, які виконують особливо часто використовувані команди, такі як Save, Open, Undo, Copy, Paste та ін. Користувач має можливість самостійно налагоджувати цю панель.

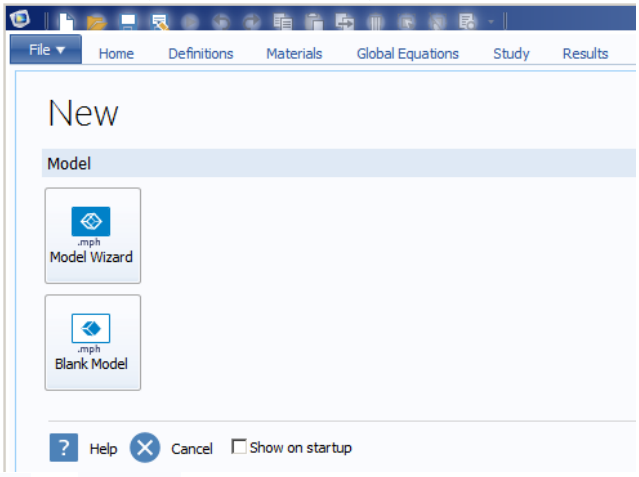
Стрічка інструментальних панелей, яка розташована вгорі, містить закладки з кнопками, які виконують більшість команд моделювання.

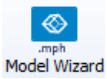
Панель конструктора моделей (Model Builder) надає графічне відображення процесу моделювання у вигляді дерева, елементи якого керують всім процесом постановки і розв'язання задачі. Кожний аспект задачі репрезентується у вигляді гілок та вузлів цього дерева. Через їх властивості можна задавати форму об'єктів, рівняння, граничні і початкові умови, будувати скінченно-елементну сітку, змінювати параметри солвера (метода розв'язання), відображати результати обчислювання в текстовому або графічному вигляді. Гілки можуть мати відгалуження та вузли, які уточнюють або розширюють можливості батьківських вузлів. Клацання правою кнопкою миші по елементу дерева моделі відкриває контекстне меню, команди якого дозволяють додати новий вузол, видалити поточний, змінити налаштування, які пов'язані з цим вузлом, і т.д. Виділення вузла в дереві приводить до відкриття панелі властивостей/налаштувань (Settings), зв'язаної з поточним вузлом, і яка дозволяє вводити властивості відповідного елементу моделі. Ми будемо називати її панеллю властивостей (ПВ) або вікном властивостей (ВВ). Введення вихідних даних з використанням цього дерева в термінології розробників COMSOL Multiphysics називається моделюванням задачі.

В графічному вікні створюються креслення областей/тіл і будуються графіки.

В інформаційному вікні відображаються числові значення розв'язків задач, а також повідомлення , які з'являються в процесі розв'язання.

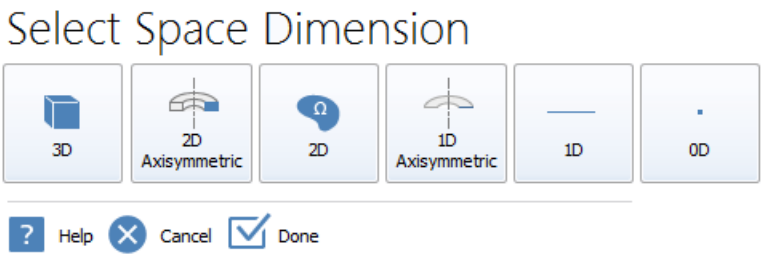
Робота в пакеті починається зі створення шаблону моделі задачі. Потрібно вибрати пункт меню File->New і у відчиненому вікні вибрати одну з двох можливостей – створення моделі за допомогою майстра (Model Wizard) або створення порожньої моделі (Blank Model).



Обидві кнопки 

і

мають однакові функціональні можливості, але перша приводить до здійснення кроків моделювання в певній послідовності. Натиснемо кнопку Model Wizard. Відчиняється вікно, в якому обирається просторова модель задачі.



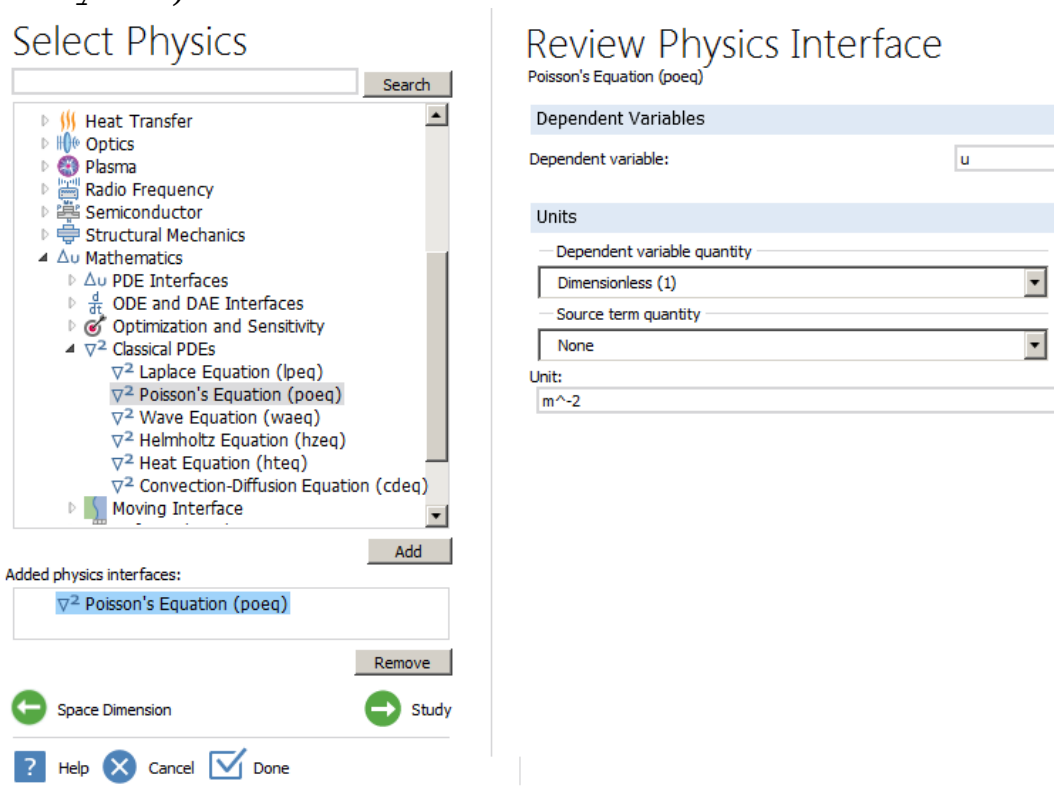
З написів на кнопках зрозуміло, яка модель буде досліджуватися. Варіант `0D` свідчить про створення просторово-незалежної моделі. Це означає, що будуть визначатися значення фізичних величин в точках. Інші варіанти вказують на створення моделі, в якій шукані величини будуть залежати від просторових координат. При цьому вони (величини) можуть не залежати від часу (стаціонарна модель) або бути функціями часу (динамічна модель). Обрання стаціонарної або динамічної моделі виконується на одному з наступних кроків роботи майстра моделей.

Припустимо, що ми бажаємо розв'язати рівняння Пуассона з граничними умовами Діріхле

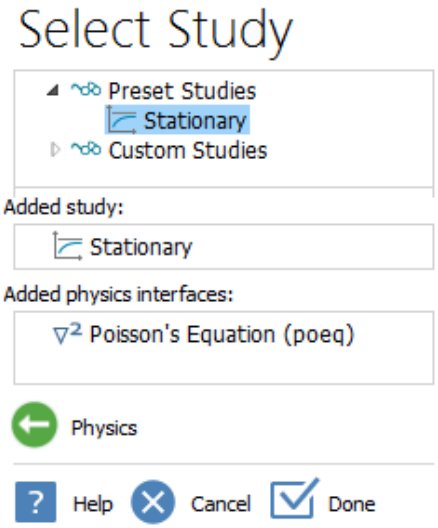
$$\Delta u = 16 \cdot \left(x^2 + y^2\right), \quad u\big|_{\partial\Omega} = 1, \tag{1}$$

де область Ω зображає круг одиничного радіуса з центром в початку координат. Вибір цієї задачі пояснюється тим, що її точний розв'язок відомий $u = \left(x^2 + y^2\right)^2$, і ми зможемо порівняти його з розв'язком, отриманим за допомогою COMSOL Multiphysics.

У вікні `Select Space Dimension` обираємо варіант `2D`. Для цього натискаємо на відповідну кнопку. Відчиняється вікно вибору фізичної моделі (`Select Physics`).



Воно містить список визначених інтерфейсів (прикладних режимів). Розкриваємо гілку Mathematics підгілку Classical PDEs та обираємо Poisson's Equation(poeq). Натискаємо кнопку Add. В поле **Added physics interfaces**, яке розташовано нижче, додається назва режиму. Праворуч в панелі Review Physics Interface в полі **Dependent Variable** можна змінити ім'я шуканої функції. Залишимо літеру u. Нижче в розділі **Units** в полі **Dependent variable quantity** обираємо Dimensionless (безрозмірний), тобто величина u буде безрозмірна. Натискаємо внизу кнопку Study. Відкривається вікно Select Study. Вибираємо варіант Stationary, який означає, що шукані функції моделі не залежать від часу (в рівняння Пуассона час не входить).



Завершуємо все натисканням на кнопку Done, розташовану внизу панелі Select Study. В результаті створюється вікно задачі з панеллю Model Builder в його лівій частині. Панель містить дерево, елементи якого керують процесом постановки і розв'язання задачі. Ми будемо називати його деревом моделі, а введення вхідних даних з використанням цього дерева будемо називати моделюванням задачі. Моделювання починається з налаштування параметрів вузлів цього дерева і, якщо потрібно, в додаванні нових вузлів.

Приклад 1. Продемонструємо процес моделювання на прикладі задачі (1).

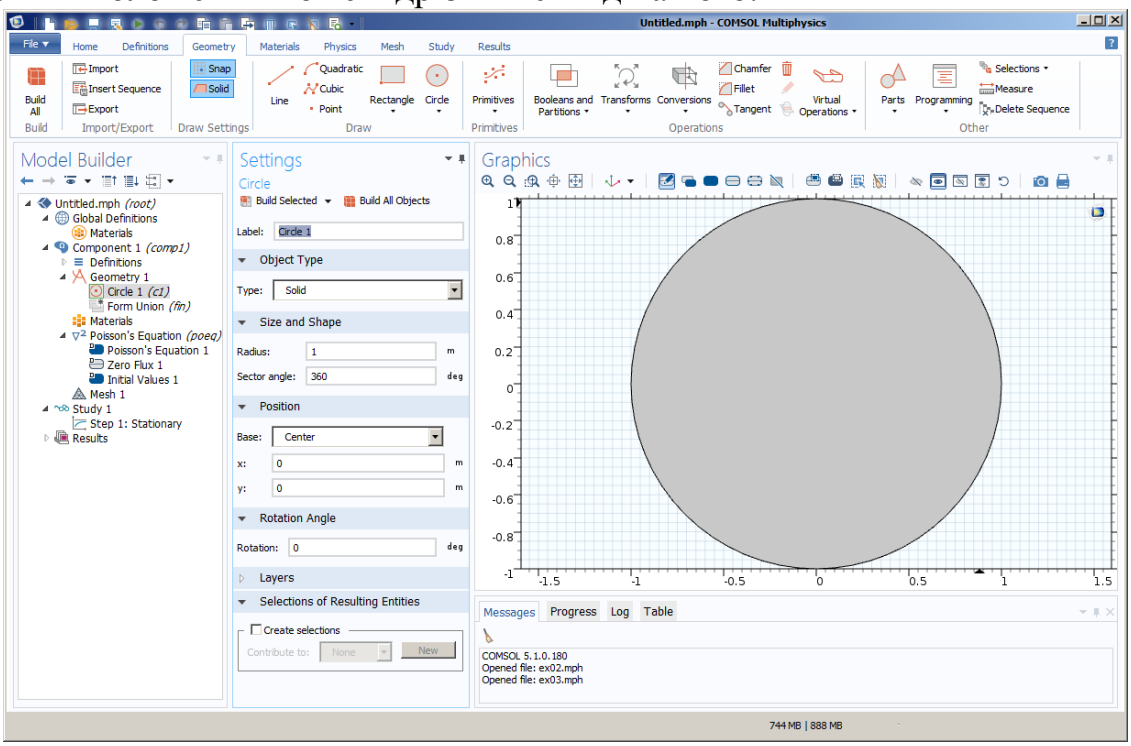
Введення умов здійснюється в чотири етапи:

1. Задання геометрії області Ω.
2. Побудова скінченно-елементної сітки.
3. Задання параметрів рівняння.
4. Задання граничних умов.

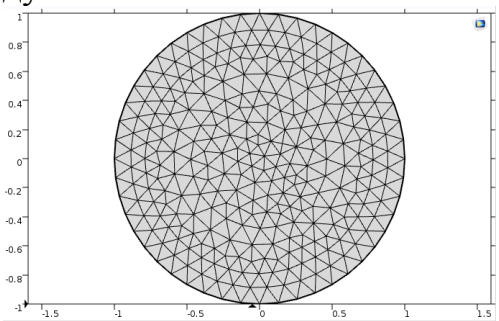
Якщо в панелі Select Physics буде обрана фізична модель не з гілки Mathematics, то можуть знадобитися інші етапи, наприклад, вибір матеріалу конструкції.

1. *Створення двовимірної області* Ω. В дереві моделі виділіть вузол **Geometry 1**. Він використовується для креслення геометрії моделі. Після виділення цього вузла графічне вікно зміниться і набуде вигляду, який налаштований для креслення плоских геометричних фігур. На стрічці панелей перейдіть до закладки Geometry. Вона містить інструменти креслення.

Клацніть на ній один раз мишею по кнопці Circle, наведіть курсор миші в область малювання, натисніть ліву кнопку, посуньте мишу і відпустіть кнопку. В результаті буде нарисований круг (по центру і радіусу). Вам не обов'язково креслити коло точно такого розміру і положення, як треба. Параметри кола можна скорегувати наступним чином. В дереві моделі в гілці **Geometry 1** з'явиться дочірній вузол **Circle 1**. Поки він активний (виділений) праворуч від нього буде відкрита панель властивостей цього вузла (нагадаємо, що цю панель ми коротко будемо позначати ПВ). В ній в розділі **Position** ви можете змінити координати центру x, y, а в розділі **Size and Shape** можете скорегувати радіус. Змініть значення координат центру так, щоб вони були рівні $x=0, y=0,$ а радіус дорівнював 1. Вгорі панелі Setting Circle натисніть кнопку Build All Objects. В результаті вікно програми набуде вигляду, показаному на наступному рисунку. Оскільки всі вікна плаваючі, то у вашій системі їх положення може відрізнятися від нашого.



2. *Побудова скінченно-елементної сітки*. В дереві моделі є вузол **Mesh 1**. Виділіть його і на стрічці панелей перейдіть на закладку Mesh. Клацніть мишею по кнопці Build Mesh. Буде виконана тріангуляція області і графічне вікно набуде наступного вигляду



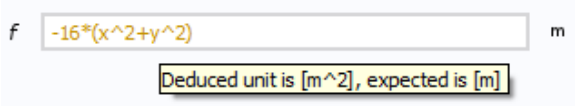
Закладка `Mesh` містить команди, які дозволяють вносити зміни в скінченно-елементну сітку. Зокрема, кнопка цієї панелі

містить випадний список команд, які дозволяють модифікувати сітку. У нашому прикладі можна погодитися з розбиттям області за замовчуванням, і сітку не змінювати.

3. *Задання параметрів рівняння*. В дереві моделі виділіть вузол **Poisson's Equathion 1**. Праворуч відкриється панель `Setting Poisson's Equathion`. В розділі **`Equation`** цієї панелі наведено рівняння Пуассона. В нашому випадку воно має вигляд $\nabla\cdot(-c\nabla u)=f$, де $\nabla=\left[\frac{\partial}{\partial x},\frac{\partial}{\partial y}\right]$ – оператор градієнта, а точка позначає скалярний добуток операторів. В координатній формі рівняння має наступний вигляд:

$$-\frac{\partial}{\partial x}\left(c\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(c\frac{\partial u}{\partial y}\right)=f\,.$$

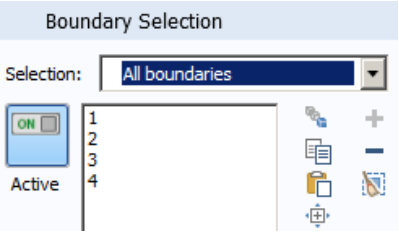
Нижче в розділі **`Diffusion Coefficient`** залиште 1 в якості значення коефіцієнта `c` цього рівняння, а в розділі **`Source Term`** введіть вираз для `f` – правої частини рівняння. Синтаксис запису математичних виразів близький до синтаксису MatLab. Вираз повинен мати вигляд $-16*(x\text{^}2+y\text{^}2)$ (з мінусом, оскільки мінус використовується в лівій частині рівняння). Зверніть увагу, що колір формули став помаранчевим. Це є попередженням, що можливо не всі дані введені коректно. В нашому випадку це пов'язано з невідповідністю фізичних одиниць вимірювання. Щоб зрозуміти, що відбувається, слід підвести курсор миші до формули. Спливе пояснююча підказка.



Зараз нас одиниці виміру не цікавлять, тому проігноруймо попередження. Якби ви ввели формулу невірно, то колір став би червоним (ознака помилки). Тоді обов'язково слід звернути увагу на спливаюче повідомлення і виправити помилку.

Зауваження. Для відключення контролю одиниць вимірювання треба у вузлі **Component 1** в списку **Unit system** вибрати `None`.

4. *Задання граничної умови*. Клацніть в дереві моделі правою кнопкою миші по гілці **Poisson's Equation (poeq)** (не по вузлу рівняння, а по імені його батьківської гілки) і в контекстному меню оберіть Dirichlet Boundary Condition. В дереві моделі з'явиться вузол **Dirichlet Boundary Condition 1**, а праворуч відкриється його панель властивостей (ПВ). У графічному вікні межа кола розбита на 4 частини. Послідовно виділіть всі частини, клацаючи по ним мишею (при цьому дуги змінюють товщину і колір). Можна також в розділі **`Boundary Selection`** в списку **Selection** вибрати `All boundaries`. Номери ділянок окружності, на яких буде задана умова Діріхле, з'являться в секції вибору нижче списку **Selection**.



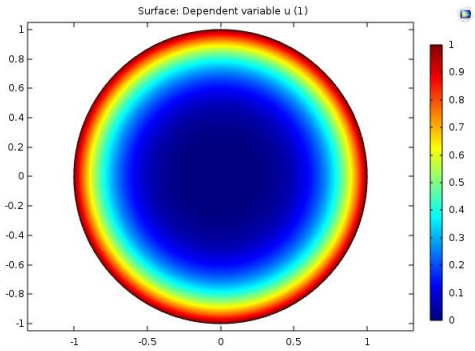
В розділі **Dirichlet Boundary Condition** цієї панелі в полі r виправіть значення 0 на 1. Цим задається гранична умова $u|_{\partial\Omega} = 1$.

Зараз все готово для розв'язання задачі. В дереві моделі правою кнопкою миші клацніть вузол **Study 1** і в випадному меню виберіть команду Compute. В правому нижньому кутку головного вікна ви побачите індикатор виконання (Progress Bar).

Команду Compute можна також виконати натисканням на клавішу F8 або клацанням миші по кнопці

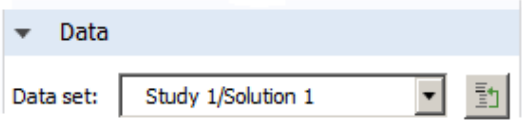
, яка розташована на закладці Home стрічки панелей. Якщо всі умови задачі введені коректно, то в графічній області ви побачите кругову зону, на якій колір точок відповідає значенню/величині розв'язку $u(x, y)$. Таке функціональне забарвлення області ми іноді будемо називати залитим (або суцільним) контурним графіком.

Зверніть увагу, що малюнок теж належить дереву моделі, а саме гілці **Results**. До обчислення ця гілка містила порожні гілки. Тепер вона виглядає наступним чином



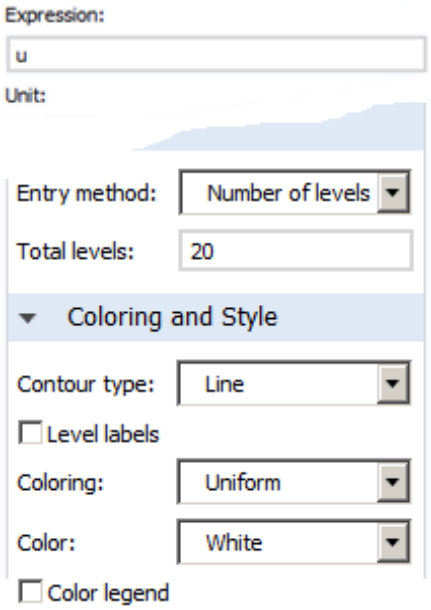
З'явився вузол **Study 1/Solution 1**, що представляє дані обчислення, і гілка **2D Plot Group 1** з вузлом **Surface 1**. Саме вузлу **Surface 1** належить побудований графік. Якщо ви виділите який-небудь інший вузол дерева моделі, наприклад, вузол **Geometry**, то вміст графічного вікна зміниться. Повторне виділення вузла **Surface 1** знову відобразить залитий контурний графік в графічному вікні.

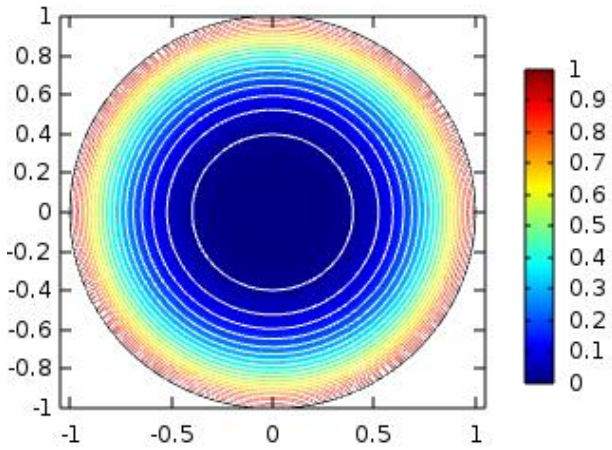
Гілка типу **2D Plot Group** призначена для побудови двовимірних графіків, а її вікно властивостей містить про них загальні відомості. Зокрема, в полі **Data Set** вказується джерело даних, яке використовується для побудови графіка.



Дані повинні «бути присутніми» в підгілці **Data Sets** гілки **Results**. В нашому випадку вони отримані в результаті обчислень і знаходяться у вузлі **Results - Data Sets – Study 1/Solution 1**. Сама гілка **2D Plot Group 1** повинна містити хоча б один дочірній вузол, який представляє графік конкретного типу. Зокрема, вузол типу **Surface** зображує залитий контурний графік, показаний вище.

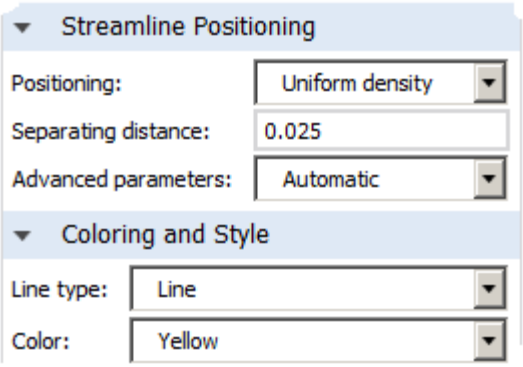
Графіки можуть накладатися один на одного. Для цього в гілку **2D Plot Group 1** можна додавати інші графічні вузли. Наприклад, клацніть по вузлу **2D Plot Group 1** правою кнопкою миші і в контекстному меню виберіть команду Contour. До групи графіків **2D Plot Group 1** додасться вузол **Contour 1**, який буде представляти графік ліній рівня виразу, записаного в полі **Expression** вікна властивостей цього вузла (в нашому випадку це `u` - ім'я розв'язку). В цьому вікні в розділі **`Levels`** в полі **Total Levels** можна задати кількість ліній рівня. В списку **Contour Type** виберіть тип графіка – `Line`, що означає зображення контурів лініями, а нижче в списку **Color** задайте колір ліній рівня, наприклад, білий. В цій **п**анелі **в**ластивостей (ПВ) також зніміть перемикач **Color legend** для того, щоб в графічному вікні не відображалася друга шкала кольорів. Частина ПВ вузла **Contour 1**, налаштованого таким чином, показана на наступному рисунку ліворуч.



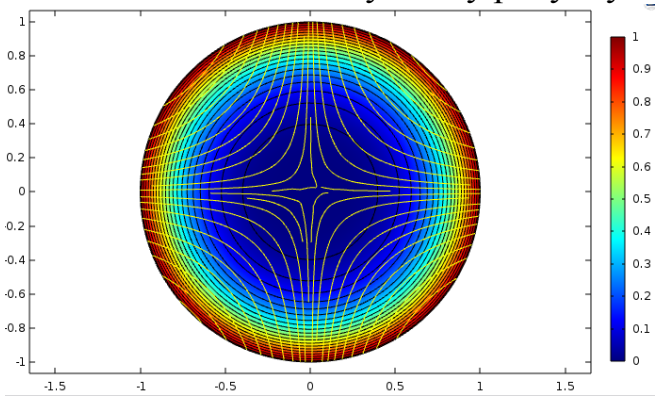


Для побудови графіка натисніть кнопку Plot, яка знаходиться вгорі панелі `Setting Contour`. В результаті, другий графік накладається на перший (див. попередній рисунок праворуч).

Додамо в цю групу графік ліній течії векторного поля $u'_x, -u'_y$. Для цього клацніть правою кнопкою миші по гілці **2D Plot Group 1** і в контекстному меню оберіть StreamLine. У вікні властивостей створеного вузла **StreamLine 1** в розділі **`Expression`** в полях **x component** і **y component** введіть `ux` і `-uy` (частинні похідні розв'язку `u` по `x` і по `y` позначаються `ux` і `uy`).

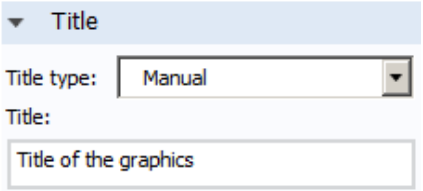


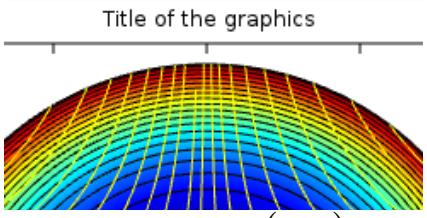
В розділі **`Streamline Positioning`** в списку **Positioning** виберіть `Uniform density` і в полі **Separating distance** задайте відстань між лініями. В розділі **`Coloring and Style`** в списку **Color** виберіть колір векторних ліній (див. попередній рисунок праворуч). Для побудови графіка натисніть кнопку Plot вгорі панелі `Setting StreamLine`. Третій графік буде накладено на перші два. Результуюче зображення наведено на наступному рисунку.



Таким чином, кожний 2D графік зображає вираз, записаний в його полі **Expression**, і графіки однієї групи накладаються в графічному вікні один на одного.

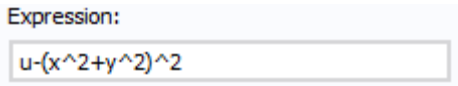
Зверніть увагу, що в вікнах властивостей є поля **Label**, в яких вводять назву вузлів дерева моделі (можна використовувати національні шрифти). Наприклад, в поле **Label** вузла **Surface 1** введіть текст `Поверхня`. В поле **Description** (якщо встановлено його перемикач) вводиться назва графіка. Вона відображається вгорі графічного вікна (дозволяється тільки латинський шрифт). В розділі **`Title`** в списку **Title type** вводиться назва заголовка, яка стоїть перед основним текстом. Якщо накладається декілька графіків, то заголовок краще задавати у вікні властивостей графічної групи **2D Plot Group 1** в розділі **`Title`**. Для цього в списку **Title type** треба вибрати `Manual` і в текстовому полі **Title** ввести заголовок об'єднаного графіка.



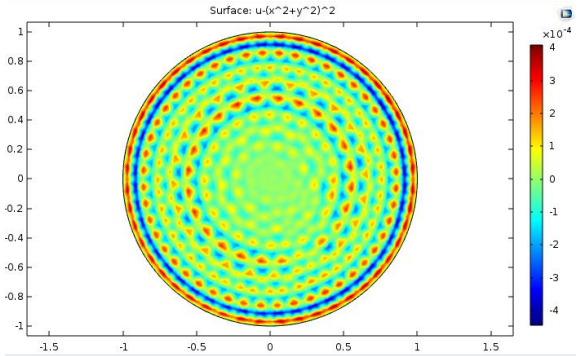


Порівняємо отриманий наближений розв'язок $u(x,y)$ з точним $w(x,y)=(x^2+y^2)^2$. Для цього побудуємо контурний графік різниці

наближеного і точного розв'язків $u(x,y)-w(x,y)$. В дереві моделі клацніть правою кнопкою миші по гілці **Results** і в контекстному меню оберіть команду 2D Plot Group. З'явиться новий вузол **2D Plot Group 2**. Додайте в нього підвузол **Surface 1** так само, як ви це робили раніше (клацанням правою кнопкою миші і вибором з контекстного меню команди **Surface**). У панелі властивостей (ПВ) цього вузла в поле **Expression** введіть вираз $\mathrm{u}-(\mathrm{x}\wedge 2+\mathrm{y}\wedge 2)\wedge 2$.



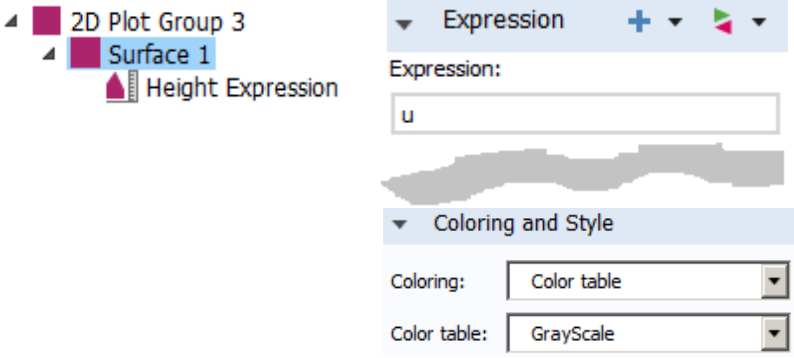
Побудуйте графік цього виразу клацанням по кнопці Plot в панелі властивостей (ПВ) вузла **Surface 1**.



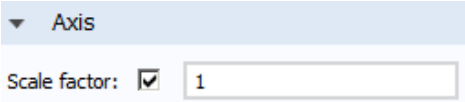
По шкалі кольорів, розташованій праворуч від рисунка, видно, що значення виразу $u-(x\wedge 2+y\wedge 2)\wedge 2$ змінюється в діапазоні від $-4\cdot 10^{-4}$ до $+4\cdot 10^{-4}$. Це означає, що абсолютна похибка (різниці наближеного і точного розв'язків) не перевищує $4\cdot 10^{-4}$.

Якщо ви побажаєте зменшити похибку, то вам слід зменшити розмір скінченних елементів сітки. Для цього треба повернутися до вузла **Mesh 1** дерева моделі і до закладки `Mesh` на стрічці. Потім слід вибрати відповідний інструмент, згустити сітку і заново виконати обчислення.

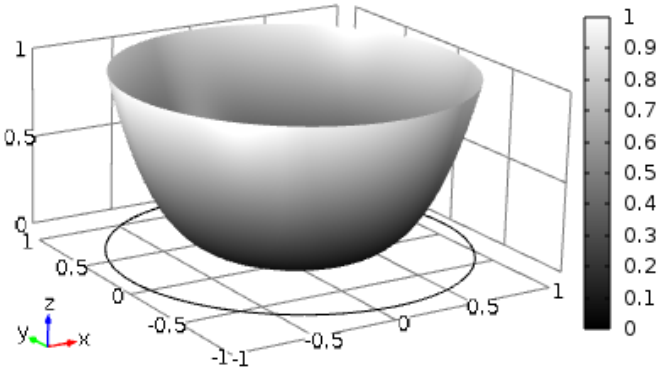
Є можливість намалювати графік розв'язку у вигляді поверхні. Для цього додайте ще одну гілку **2D Plot Group 3** і в ній підгілку **Surface 1**, в якій створіть вузол **Height Expression 1**. Він створюється так само, як і будь-який інший вузол, клацанням правої кнопки миші по батьківському вузлу і вибором в контекстному меню команди Height Expression. В ПВ вузла **Surface 1** в розділі **`Coloring and Style`** в списку **Color table** виберіть таблицю кольорів `GrayScale`.



В ПВ вузла **Height Expression 1** в розділі **`Axis`** в поле **Scale factor** встановіть одиничний масштабний коефіцієнт графіка.

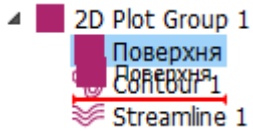


Нарешті натисніть кнопку Plot, яка будує графік поверхні функції.

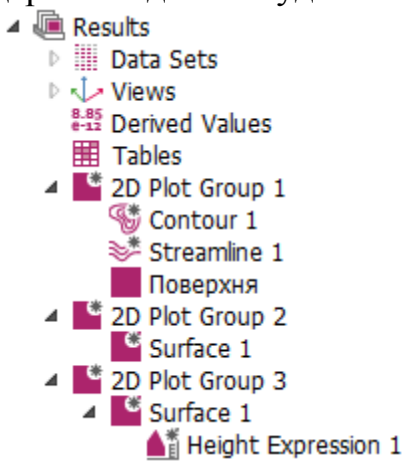


Те, що це поверхня функції `u` визначається установкою в полі **Expression** вузла **2D Plot Group 3 - Surface 1**, бо у дочірньому вузлі **Height Expression 1** в полі **Height data** встановлено `From parent`, що означає деформування поверхні по даним батьківського вузла.

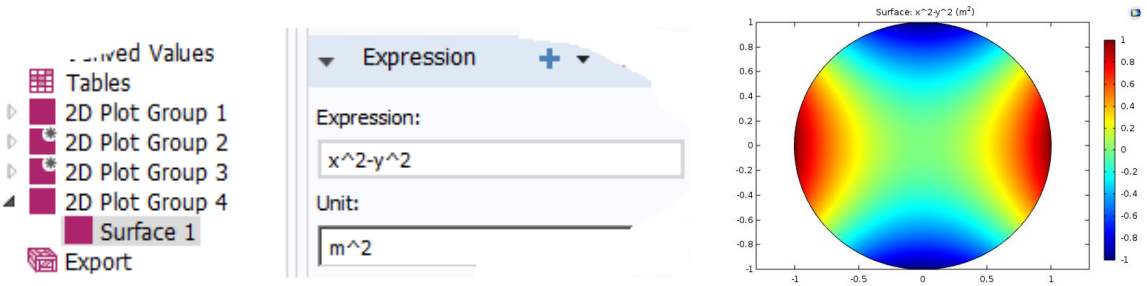
Положення вузлів в дереві моделі не фіксоване. Зокрема, графічні вузли можна переміщати в межах групи графіків або навіть переміщати в іншу графічну групу. Наприклад, захопіть лівою кнопкою миші вузол **Поверхня** (натисніть кнопку і не відпускайте) і перетягніть вузол вниз гілки **2D Plot Group 1**. В процесі переміщення положення вузла позначається червоним відрізком.



В результаті, гілка **Results** дерева моделі набуде наступного вигляду.

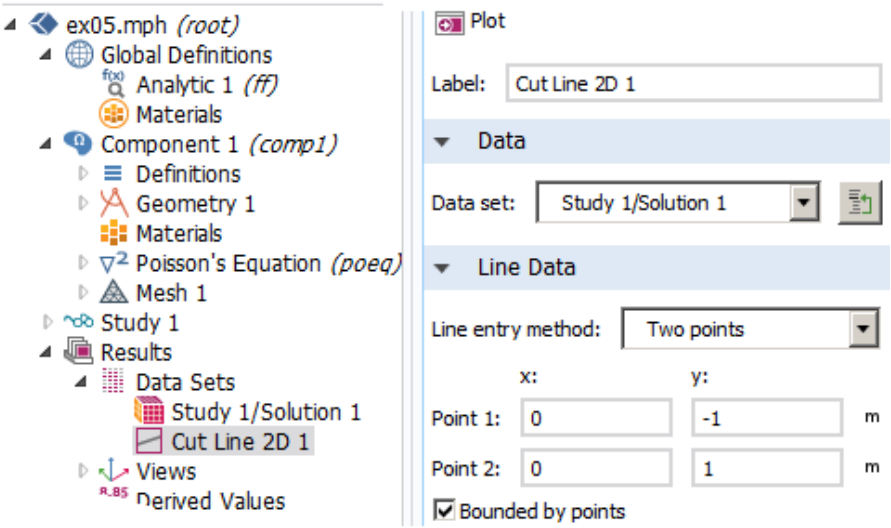


Графік може бути не пов'язаний з розв’язком `u`. Наприклад, додамо до дерева нашої моделі ще одну гілку **2D Plot Group 4** з вузлом **Surface**. В його ПВ в поле **Expression** введіть вираз `x^2-y^2` і натисніть кнопку Plot. В зоні одиничного кола буде побудовано залитий контурний графік гіперболічного параболоїда.



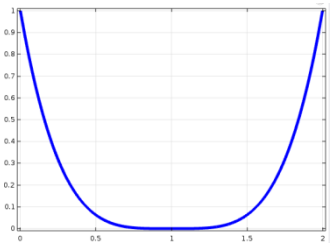
Зверніть увагу на те, що в розділі **`Data Set`** вікна властивостей вузла **2D Plot Group 4** все одно повинно бути вказано джерело даних `Study 1/Solution 1`. Воно зберігає не тільки значення шуканої функції `u`, а й координати вузлів скінченно-елементної сітки. Тому графік виразу `x^2-y^2` побудований на цій сітці (в одиничному колі).

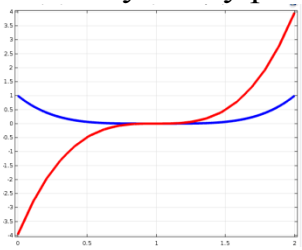
Будувати графік розв'язку `u` можна не на всій області. Наприклад, побудуємо його вздовж вертикального діаметру нашого кола. Для цього в гілці **Data Sets** треба створити інше джерело даних – множину точок, над яким буде будуватися графік. Клацніть правою кнопкою миші по гілці **Data Sets** і в контекстному меню оберіть команду Cut Line 2D (лінія перерізу двовимірної області). В його ПВ в розділі **`Data`** в списку **Data set** виберіть джерело даних `Sudy 1/Solution 1`, а в списку **Line entry method** виберіть `Two points` (переріз по двом точкам). Введіть координати точок початку і кінця лінії перерізу в полях **x, y**. Панель властивостей вузла **Cut Line 2D** заповніть так, як показано на наступному рисунку (показана частина цього вікна).



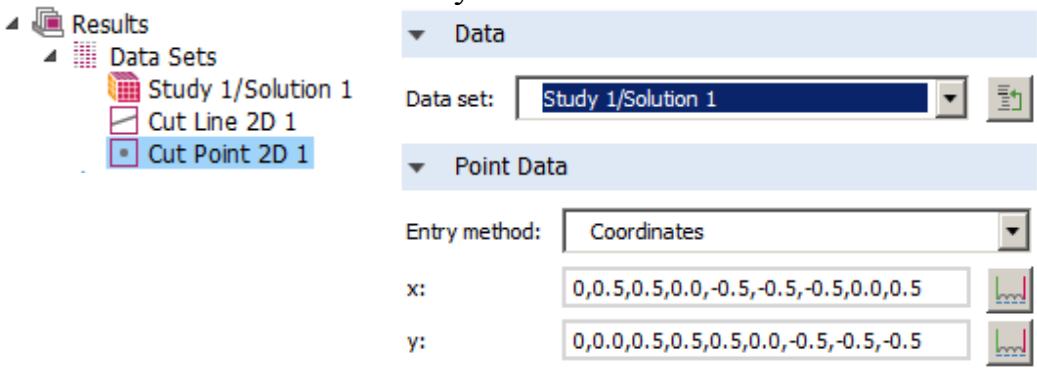
Тепер додайте графічну гілку **1D Plot Group 5**. В її ПВ в списку **Data set** виберіть `Cut Line 2D 1`. Додайте підвузол **Line Graph**. В його ПВ в списку **Data set** виберіть `From parent`, а в поле **Expression** введіть `u` (ім'я розв'язку). Зверніть увагу, що в розділі **`x-Axis Data`** в списку **Parameter** обрано значення `Arc length`, тобто наш графік віднесений до довжини дуги перерізу (діаметра). Вгорі ПВ вузла **Line Graph 1** натисніть кнопку Plot. В графічному вікні буде побудований графік, показаний на наступному рисунку ліворуч. Для побудови графіка частинної похідною розв'язку вздовж цього відрізка додайте в цю графічну групу вузол **Line Graph 2** і в його ПВ в поле **Expression** введіть `uy` (похідна розв'язку по змінній `y`). Решту полів заповніть так само, як і для

вузла **Line Graph 1**. Після натискання кнопки Plot графічне вікно оновиться і в ньому буде побудований другий графік. Результуюче графічне вікно, яке відповідає вузлу **1D Plot Group 5**, показано на наступному рисунку праворуч.

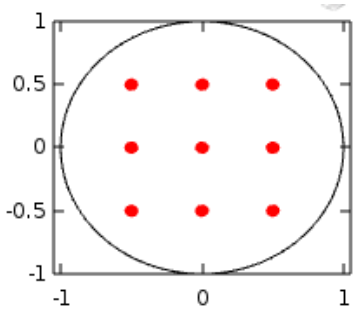




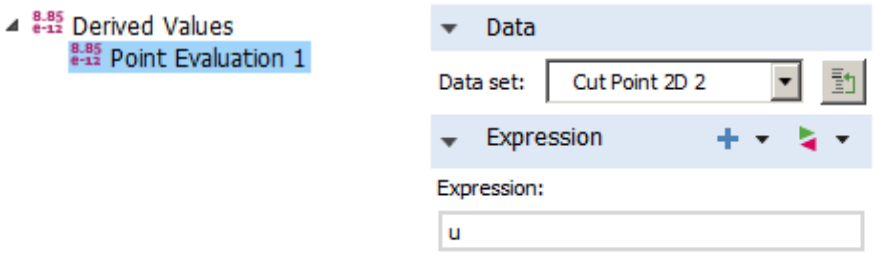
Знайдемо числові значення розв'язку в деяких точках. Для цього треба додати відповідне джерело даних в дерево моделі. Клацніть правою кнопкою миші по вузлу **Results - Data Sets** і у випадному списку виберіть команду Cut Point 2D. В гілку **Results - Data Sets** буде додано вузол **Cut Point 2D 1**. В його ПВ в розділі **`Data`** в списку **Data set** виберіть розв'язок `Sudy 1/Solution 1`, а в розділі **`Point Data`** введіть через кому *x* - координати і *y* - координати точок. Нижче показана частина ПВ вузла **Cut Point 2D 1**.



Натиснувши на кнопку Plot вгорі ПВ вузла **Cut Point 2D 1**, ви отримаєте зображення точок.



Тепер клацніть правою кнопкою миші по вузлу **Results - Derived Values** і у випадному списку виберіть команду Point Evaluation. З'явиться вузол **Results - Derived Values - Point Evaluation 1**. В його ПВ в списку **Data set** слід вибрати `Cut Point 2D 1` (створене раніше точкове джерело даних). Також в цій ПВ потрібно ввести вираз, значення якого буде обчислюватися. В нашому випадку ми будемо обчислювати значення розв'язку в точках. Тому в поле **Expression** вводимо `u` (нижче показана частина ПВ вузла **Point Evaluation 1**).



Натисніть на кнопку Evaluate вгорі ПВ вузла **Point Evaluation 1** і в правому нижньому куту програми з'явиться таблиця значень розв’язку в кожній вказаній вами точці.

Dependent variable u, Point: (0, 0)	Dependent variable u, Point: (0.5, 0)	Dependent variable u, Point: (0.5, 0.5)	Dependent variable u
-0.00000020173	0.062406	0.24998	0.062505

Створену модель можна зберегти в файл, який має розширення імені `*.mph`. Зберігається вся модель разом з результатами обчислень (якщо вони виконувалися) і графікою. Пізніше ви можете відкрити такий файл і продовжити роботу з моделлю.

▪

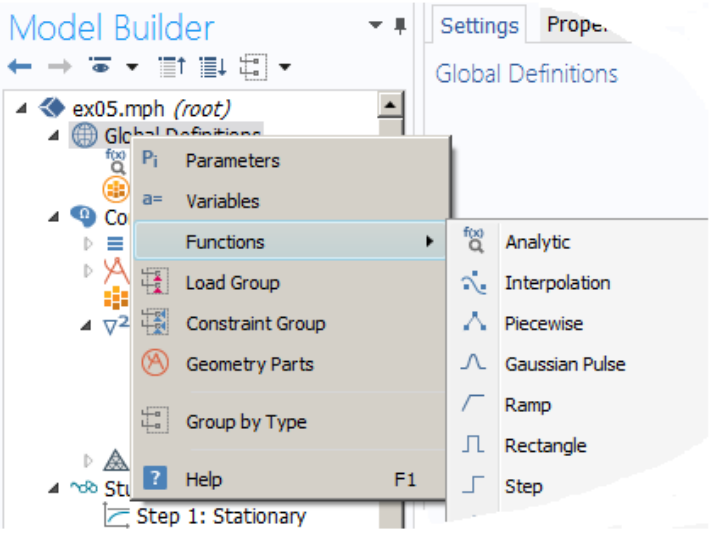
Зауваження 1. В гілці **Results - Data Sets** джерела даних типу `Cut Point` використовуються для обчислення значень виразів в точках та побудови в точках «часових» графіків; вузли типу `Cut Lines` в основному призначені для обчислення і візуалізації виразів вздовж деяких ліній, наприклад, вздовж лінії перерізу будь-якої двовимірної області. Вузли типу `Cut Plane` створюються для візуалізації розв’язків/виразів по перерізах тривимірних областей.

Зауваження 2. Отримати числове значення розв’язку в точці можна по-іншому. Активуйте графічний вузол, в якому побудований графік з функціональної забарвленням (не вузол групи, а вузол графіка), і на графіку виконайте подвійне клацання миші в околі бажаної точки. З'явиться таблиця (якщо її ще немає), в яку додасться рядок зі значеннями координат точки клацання і значенням виразу в ній. Кожне нове клацання буде додавати один рядок в цю таблицю.

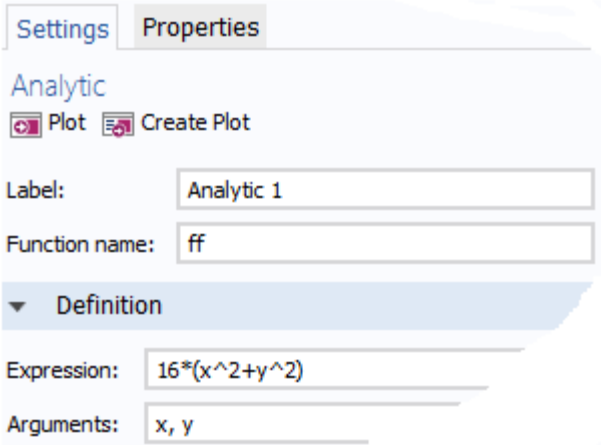
1.3. Функції і графіки.

Система COMSOL Multiphysics не використовує вбудованої мови програмування на зразок MatLab, але в ній є можливість записувати досить складні математичні вирази. Окрім того, користувач може створювати власні функції. Кожній функції в дереві моделі повинен відповідати свій вузол. Функції створюються в гілках **Global Definitions** або **Component - Definitions** дерева моделі вибором відповідного шаблону зі списку.

Припустимо, що в прикладі 1 попереднього параграфа праву частину рівняння (1) ми збираємося вводити у вигляді аналітичної функції. Для її створення клацніть правою кнопкою миші по вузлу **Global Definitions** дерева моделі і виберіть команду Functions->Analytic.



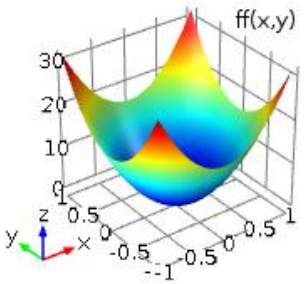
Для функції буде створено власний вузол і відкриється панель властивостей, призначена для задання функції за допомогою аналітичного виразу. В поле **Function name** введіть ім'я функції, наприклад, `ff`. В полі **Arguments** через кому перелічіть імена аргументів функції, а в полі **Expression** запишіть математичний вираз, використовуючи імена аргументів, константи і знаки математичних операцій. Заповнені поля в ПВ нашої функції показані на наступному рисунку.



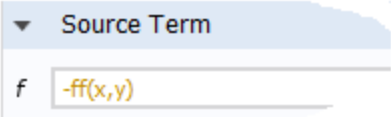
Перевірку коректності задання функції можна виконати шляхом побудови її графіка. Для цього розкрийте розділ **`Plot Parameters`** і в таблиці задайте діапазон зміни аргументів функції на графіку (це не обмежує діапазон аргументів при подальшому використанні).

Argument	Lower limit	Upper limit
x	-1	1
y	-1	1

Після цього натисніть кнопку Plot вгорі ПВ. В графічному вікні буде побудований графік нашої функції $\mathrm{ff}(x, y)$.



Перейдіть у вікно властивостей вузла **Component 1 - Poisson's Equation (poeq) - Poisson's Equation 1** і в розділі `Source Term` в поле **f** введіть $-\mathrm{ff}(x,y)$.



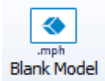
На закладці `Home` стрічки панелей натисніть кнопку = Compute і отримайте той же розв'язок, що і раніше.

В системі є шаблони для створення аналітичних, інтерполяційних, кускових, пилкоподібних, східчастих і інших функцій. Можна використовувати функції, створені в Matlab. У кожного типу функцій є свої особливості. Наприклад, Piecewise (кускові) функції допускають використання тільки одного аргументу. З особливостями створення різних функцій ми познайомимо вас у міру їх появи в наших прикладах.

Після створення функцій часто виникає потреба подальшої роботи з їх графіками (поєднання, накладання, тощо). З розібраного раніше прикладу ви бачили, що в гілці **Results** дерева моделі можна будувати і обробляти графіки розв'язків. Але якщо треба працювати лише з графіками функцій, то це можна робити без моделювання і розв'язання фізичних задач. Для цього потрібно виконати всього три кроки:

- створити функцію;
- в гілці **Results->Data Sets** створити джерело даних (`Data set`); в деякому сенсі джерело даних можна розглядати як область визначення функції, а точніше як область зміни її аргументів при побудові графіка;
- створити групу графіків **Plot Group** з графічним вузлом того чи іншого типу, наприклад, **Line Graph**; в ПВ цього вузла вказати функцію і джерело даних (`Data set`).

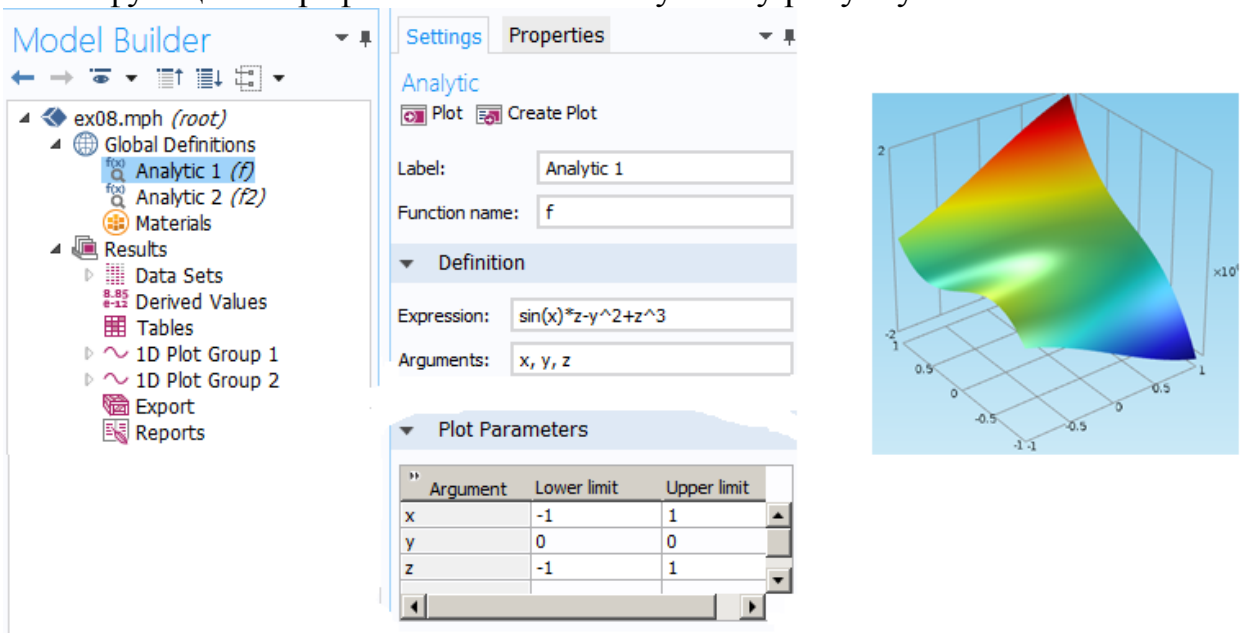
Кожен з кроків можна виконати декількома способами. Для прикладу побудуємо кілька графіків функцій.

Приклад 1. Починаємо зі створення порожньої моделі. Вибираємо пункт меню File->New і у відкрившомуся вікні вибираємо створення порожньої моделі (`Blank Model`), тобто тиснемо кнопку 

.

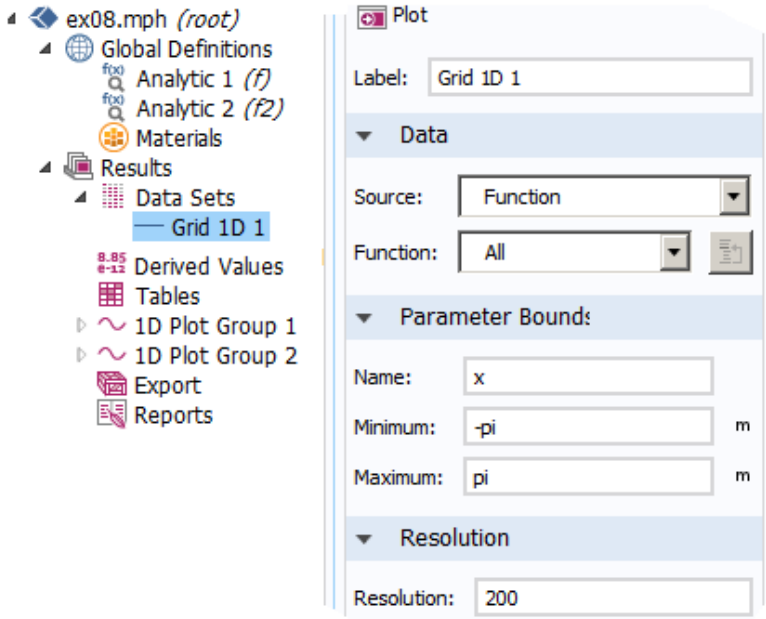
Створюємо функцію. Клацніть правою кнопкою миші по вузлу дерева моделі **Global Definitions** і виберіть команду Functions->Analytic. Для функції створиться вузол і відкриється панель властивостей, призначена для задання функції за допомогою аналітичного виразу. В поле **Function name** введіть ім'я функції, наприклад, `f`. В полі **Arguments** через кому перелічіть імена

аргументів функції `x,y,z` (функція може мати кілька аргументів). В поле **Expression** введіть математичний вираз `sin(x)*z-y^2+z^3`. Для контролю побудуйте графік функції двох змінних, задавши фіксоване значення одного з аргументів в розділі **`Plot Parameters`** (щоб присвоїти `y = 0` задайте нульові нижню і верхню межі зміни `y`), і натисніть кнопку Plot. Заповнені поля ПВ нашої функції і її графік показані на наступному рисунку.



Аналогічно створіть другу функцію однієї змінної `f2` з виразом `x^3`.

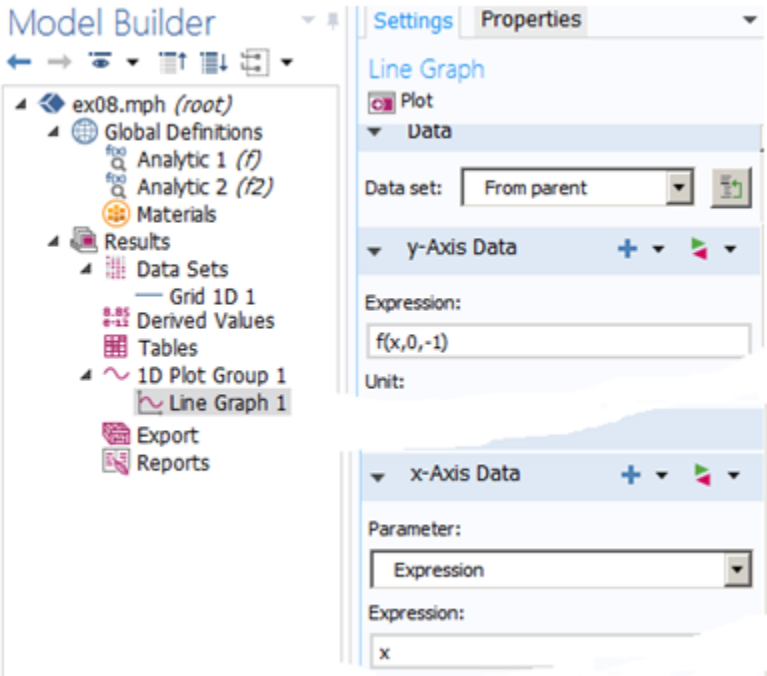
Створюємо одновимірне *джерело даних*, яке буде використовуватися для побудови графіків функцій. Для цього у випадному меню гілки **Results - Data Sets** оберіть команду More Data Sets-> Grid 1D і заповніть ПВ створеного вузла **Grid 1D** так, як показано нижче.



Вибір в списку **Source** значення `Function` означає, що на множині точок **Grid 1D** буде обчислюватися значення функції. У списку **Function** обираємо потрібну функцію (або `All`, якщо збираємося будувати кілька графіків). В полі

Resolution вказується кількість точок дискретизації відрізка, який визначається значеннями, заданими в полях **Minimum** і **Maximum**.

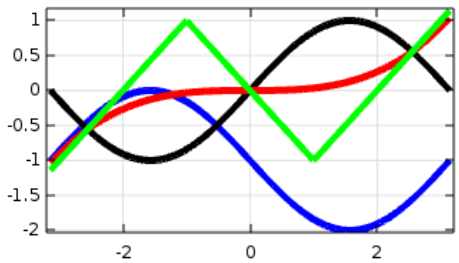
Створюємо групу графіків. Для цього в гілку **Results** додайте гілку типу **1D Plot Group** і в ній створіть вузол типу **Line Graph**. В ПВ гілки **1D Plot Group 1** в списку **Data Set** виберіть `Grid 1D 1` (одновимірна множина, на якій збираємося будувати графік). У вікні властивостей вузла **Line Graph 1** в розділі **`y-Axis Data`** в полі **Expression** введіть вираз `f(x,0,-1)`. В розділі **`x-Axis Data`** в списку **Parameter** виберіть `Expression`, і в поле **Expression** введіть `x`. В розділі **Coloring and Style** задайте стиль, колір і товщину лінійного графіка.



Для побудови графіка натисніть кнопку Plot. Діючи аналогічно, у вузол **1D Plot Group 1** додайте ще три підвузли типу **Line Graph**, в яких в полях **Expression** розділів **`y-Axis Data`** введіть вирази

```
f(x,1,1)
f2(x)/30
x-abs(x+1)+abs(x-1).
```

Решту властивостей задайте такими же, як у вузла **1D Plot Group 1**. Після побудови всіх кривих графічне вікно повинно набути наступного вигляду.



Побудуємо кілька плоских кривих по їх параметричним рівнянням.

В тій же моделі в гілку **Results** додайте вузол графічної групи **1D Plot Group 2** і створіть в ньому підвузол **Line Graph**. В ПВ гілки **1D Plot Group 2** в списку **Data Set** оберіть `Grid 1D 1`. У вікні властивостей вузла **Line Graph 1** в

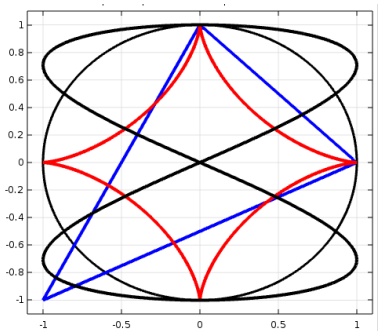
розділі **y-Axis Data** в полі **Expression** введіть вираз, наприклад, `cos(x)`. В розділі **x-Axis Data** в списку **Parameter** обиріть `Expression`, і в поле **Expression** введіть `sin(x)`. Цим ми задаємо параметричне рівняння кола. В розділі **Coloring and Style** задайте стиль, колір і товщину кривої. Для побудови кривої натисніть кнопку Plot. В цю ж гілку **1D Plot Group 2** додайте ще три підвузли типу **Line Graph**. В їх розділах **x-Axis Data** в списках **Parameter** оберіть `Expression`, і в полях **Expression** в розділах **y-Axis Data** і **x-Axis Data** введіть наступні пари виразів (параметричні рівняння кривих):

a) `abs(x)/2-3*abs(x-2)/2+abs(x-3)-1`
`abs(x)-3*abs(x-1)/2+abs(x-3)/2-1`

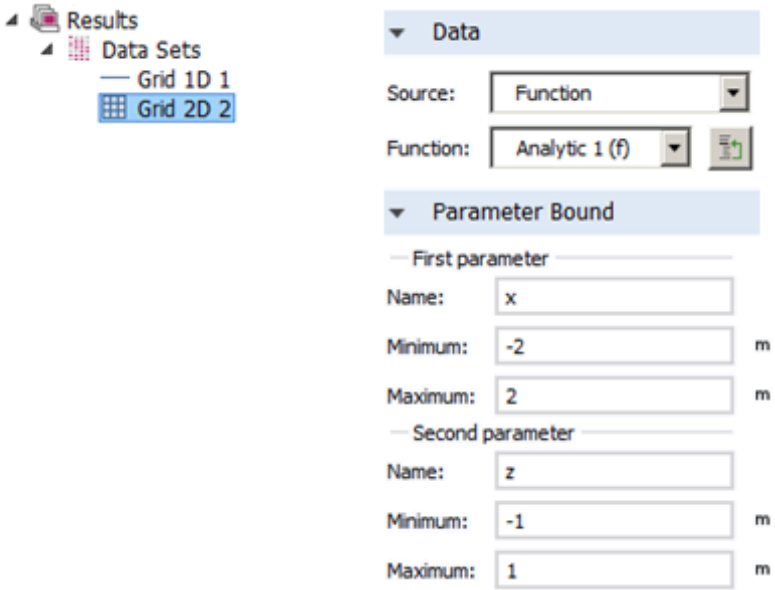
b) `cos(pi*x)^3`
`sin(pi*x)^3`

c) `cos(x)`
`sin(2*x)`

Після побудови всіх кривих кнопкою Plot графічне вікно набуде наступного вигляду.

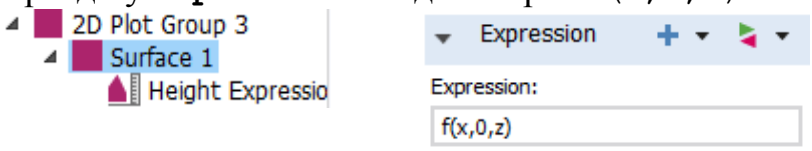


Можна будувати графіки функцій двох змінних. Для цього в гілці джерел даних **Results->Data Sets** потрібно створити двовимірну множину точок, яка буде використовуватися при побудові графіків. У випадному меню гілки **Data Sets** оберіть команду More Data Sets->Grid 2D і заповніть ПВ створеного вузла **Grid 2D 2** так, як показано нижче (не пропустіть налаштування в полі **Function**, яке повинно бути `Analitic 1(f)` або `All`).

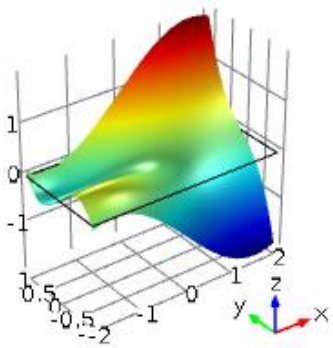


Зверніть увагу, що у другого параметра ми змінили ім'я. Для цього в розділі **`Second parameter`** в поле **name** ми ввели `z`.

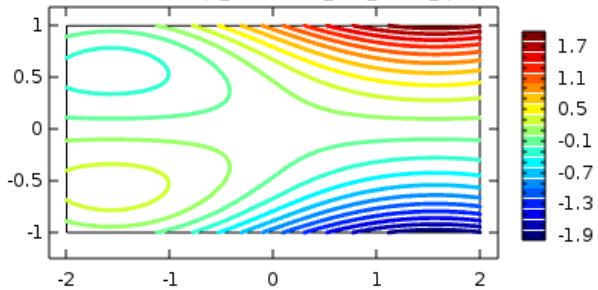
В гілку **Results** додайте гілку графічної групи **2D Plot Group** і в ній створіть вузол типу **Surface**. В ПВ створеної гілки **2D Plot Group 3** в списку **Data Set** оберіть `Grid 2D 2`. У вікні властивостей вузла **Surface 1** в полі **Expression** розділу **`Expression`** введіть вираз `f(x,0,z)`.



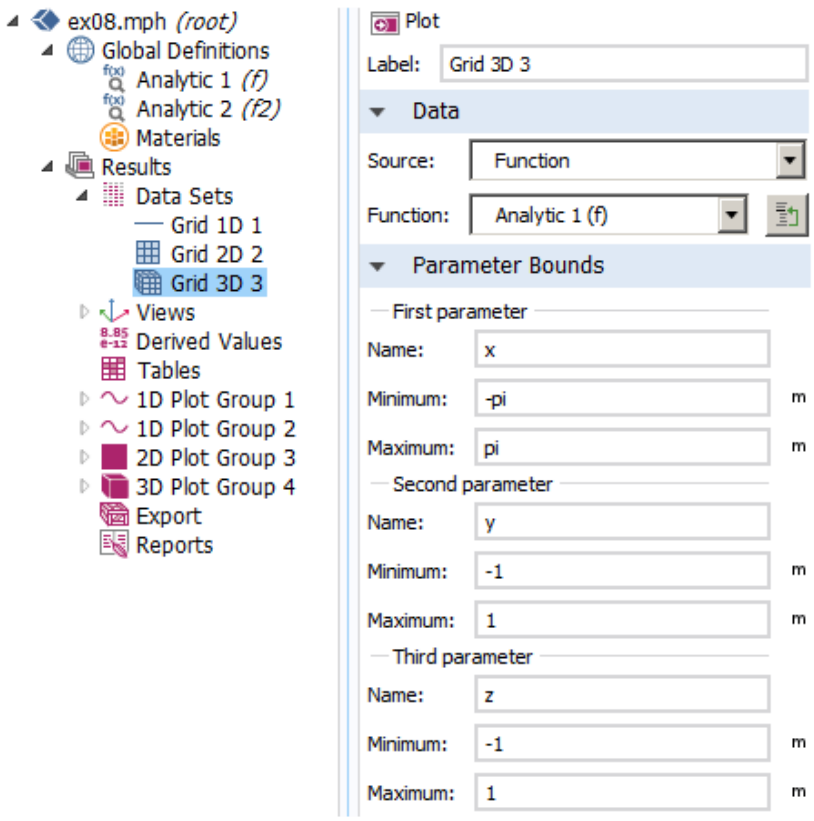
Щоб побудувати графік поверхні (а не суцільний кольоровий графік) до вузла **Surface 1** додайте підвузол **Height Expression**. В його ПВ в поле **Scale factor** (масштабний множник) введіть одиницю і натисніть кнопку Plot. Буде побудований наступний графік



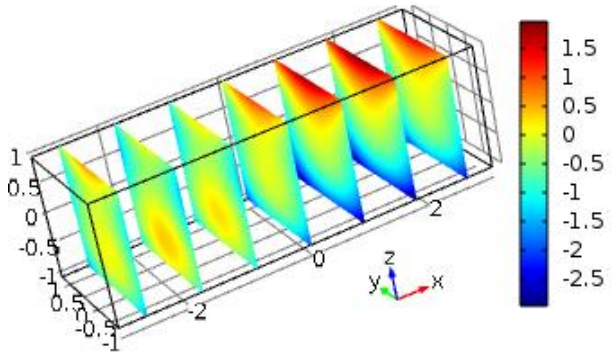
Замість вузла **Surface 1** в гілку **2D Plot Group 3** можна додати вузол **Contour 1**. Для цього клацніть правою кнопкою миші по вузлу **Surface 1** і в випадному меню виберіть команду Delete для видалення вузла або виберіть команду Disable для його деактивації, а потім додайте вузол типу **Contour**. В ПВ вузла **Contour 1** в поле **Expression** розділу **`Expression`** введіть попередній вираз `f(x,0,z)`, в поле **Total levels** розділу **`Levels`** введіть кількість ліній `20`, а в списку **Contour type** розділу **`Coloring and Style`** виберіть `Line`. В результаті буде побудований контурний графік функції 2 змінних.



Для графічного зображення функції 3 змінних можна використовувати «пошаровий» (slice) графік (графік перерізів). Для цього в гілці **Data Sets** командою випадного меню More Data Sets->Grid 3D створіть тривимірну множину точок **Grid 3D** (точніше вузол в дереві моделі). Заповніть його ПВ так, як показано на наступному рисунку.

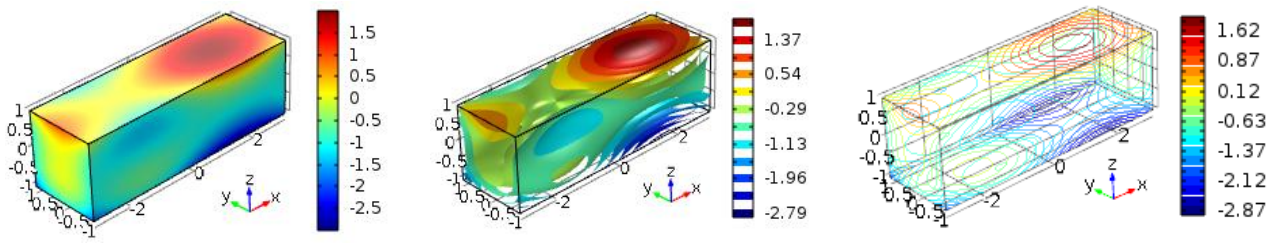


В гілку **Results** додайте підгілку **3D Plot Group** і в ній створіть вузол типу **Slice**. В ПВ гілки **3D Plot Group 4** в списку **Data Set** виберіть `Grid 3D 3`. У вікні властивостей вузла **Slice 1** в поле **Expression** розділу **`Expression`** введіть вираз `f(x,y,z)`. В розділі **`Plane Data`** в полі **Planes** введіть кількість перерізів/шарів, наприклад, 7. Натисніть кнопку Plot. В результаті буде побудовано наступне зображення.



Колір точок, розташованих на зрізах, вказує значення функції. Шкала кольорів, яка представляє значення функції, показана праворуч від графіка.

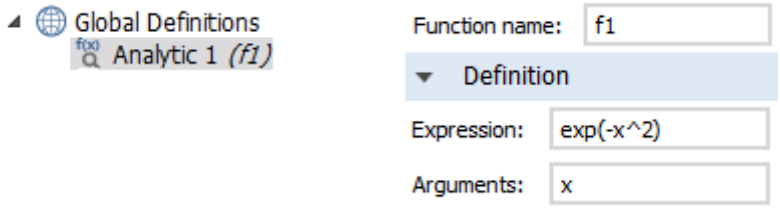
Ту ж функцію 3 змінних можна зобразити по-іншому. Наприклад, створіть ще графічні групи і вузли в них: **3D Plot Group 5 - Surface**, **3D Plot Group 6 - Isosurface**, **3D Plot Group 7 - Contour**. На наступному рисунку показані зображення функції `f(x,y,z)` побудовані на цих графіках (зліва направо **Surface**, **Isosurface** і **Contour**). Вікна властивостей цих вузлів мають поле **Expression** для введення виразу `f(x,y,z)`, а ПВ двох останніх вузлів мають ще поле **Total Levels** для завдання на графіку кількості поверхонь/ліній рівня.



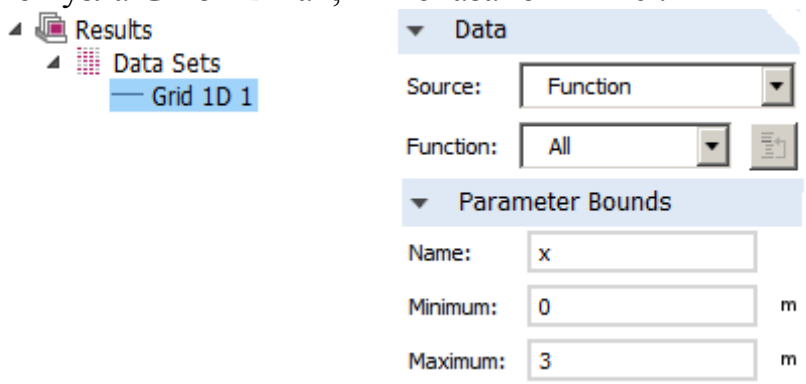
■

Розв’язок і будь-які функції можна інтегрувати.

Приклад 2. Побудуємо графік функції $f(x)=\int_0^x e^{-\tau^2}d\tau$. Для цього створимо порожній проект. У ньому створимо аналітичну функцію $f_1(x)=\exp(-x^2)$.



Далі створимо одновимірне джерело даних. Для цього у випадному меню гілки **Results - Data Sets** обираємо команду More Data Sets->Grid 1D і заповнюємо ПВ створеного вузла **Grid 1D** так, як показано нижче .



Створюємо групу графіків **1D Plot Group** і в ній створюємо вузол типу **Line Graph**. В ПВ гілки **1D Plot Group 1** в списку **Data Set** вибираємо `Grid 1D 1`. У вікні властивостей вузла **Line Graph 1** в розділі **`y-Axis Data`** в полі **Expression** вводимо вираз `integrate(f1(v),v,0,x)`.

Оператор `integrate` має наступний синтаксис

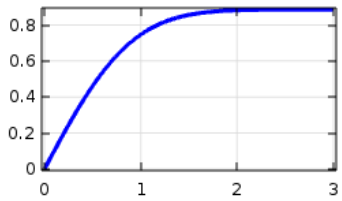
`integrate(expr(v),v,v1,v2)`,

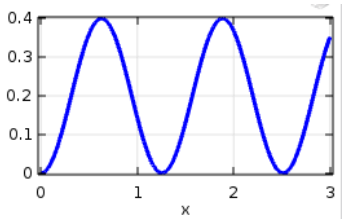
де `expr(v)` являє підінтегральний вираз, `v` – ім'я змінної інтегрування, `v1` і `v2` – нижня і верхня межа інтегрування, які не зобов'язані бути константами.

В розділі **`x-Axis Data`** в списку **Parameter** вибираємо `Expression`, і в поле **Expression** вводимо `x`. В розділі **`Coloring and Style`** задаємо стиль, колір і товщину лінії. Натискаємо кнопку Plot і отримуємо графік, показаний на наступному рисунку зліва.

Інтегрувати можна будь-який вираз, а не тільки створену заздалегідь функцію. Наприклад, створіть ще одну графічну гілку **1D Plot Group 2** з вузлом **Line Graph 1**. В ПВ гілки **1D Plot Group 2** в списку **Data set** виберіть `Grid 1D 1`. В ПВ його підвузла **Line Graph 1** в розділі **`y - Axis Data`** в полі

Expression введіть вираз `integrate(sin(v*5),v,0,x)`, а в розділі **`x - Axis Data`** в полі **Expression** введіть `x`. Натискання на кнопку Plot побудує графік функції $\int_0^x \sin(5v)\,dv$, який показано на наступному рисунку праворуч.



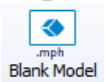


Інтегрування виразів, які залежать від часу, виконується оператором `timeint`.

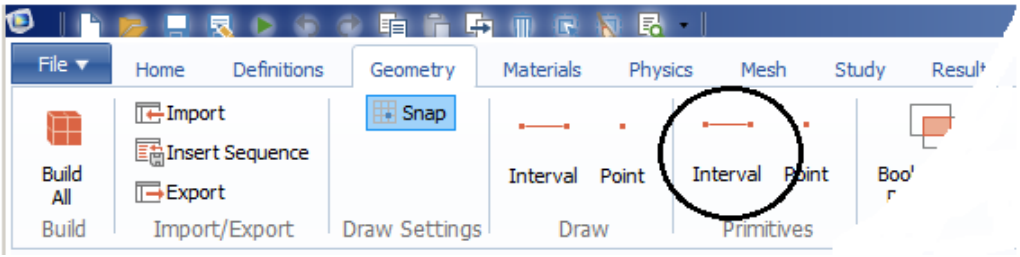
Крім використання операторів `integrate` і `timeint` існують інші способи обчислення інтегралів. Ви познайомитеся з ними пізніше.

■

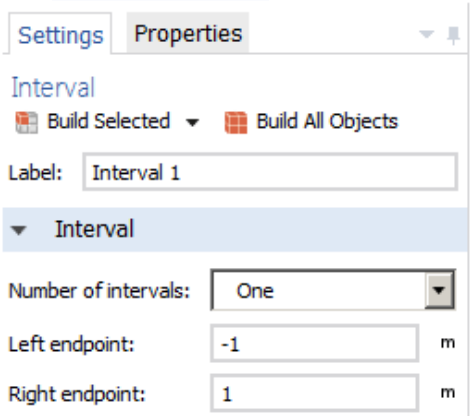
Інший спосіб задання множини точок, на якій будується графік функції, полягає в тому, щоб використовувати скінченно-елементну сітку, створену при моделюванні задачі, або створену спеціально.

Приклад 3. Створимо порожню модель. Виберіть пункт меню File->New і у відкрившомуся вікні виберіть створення порожньої моделі (`Blank Model`), тобто натисніть кнопку 

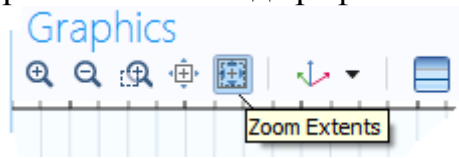
. Клацніть правою кнопкою миші по найвищому вузлу дерева моделі (**Untitled.mph**) і в випадному меню виберіть команду Add Component->1D. Цим ми створюємо одновимірну модель. З'являється вузол **Component 1** і кілька його підвузлів, з яких нас цікавлять вузли **Geometry 1** і **Mesh 1**. Виділіть вузол **Geometry 1** і на стрічці панелей відкрийте закладку `Geometry`, на якій клацніть по кнопці Interval в розділі **`Primitives`**.



В гілку **Geometry 1** додається вузол **Interval 1**. В його ПВ задайте координати лівої і правої кінцевих точок відрізка, наприклад, `-1` і `1`. Щоб намалювати відрізок, клацніть по кнопці Build All Objects вгорі ПВ вузла **Interval 1**.

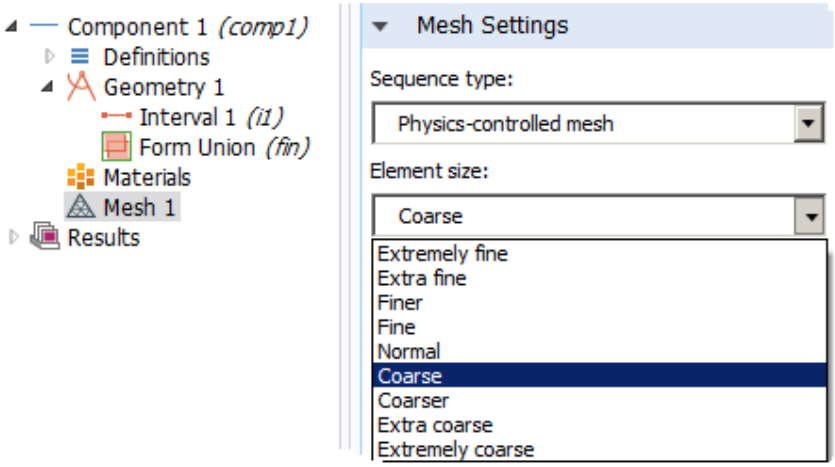


Якщо зображення відрізка вийшло за межі графічного вікна, то натисніть на кнопку Zoom Extents, яка розташована над графічним вікном.

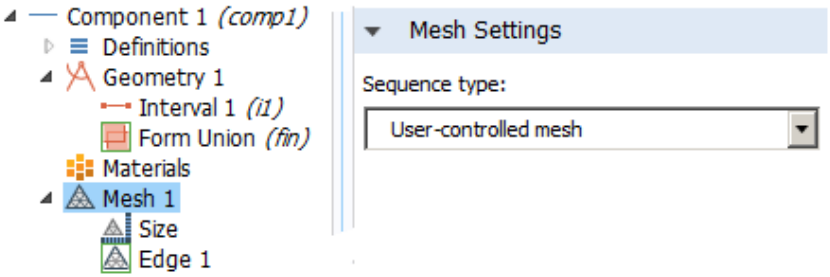


Тепер побудуйте скінченно-елементну сітку на цьому відрізку. Для цього виділіть вузол **Mesh 1** і на стрічці панелей, на закладці `Mesh` натисніть кнопку Build Mesh. У графічному вікні зображення відрізка набуде наступного вигляду.

Кількість точок розбиття визначається довжиною скінченних елементів. Їх зумовлені розміри вибираються в списку **Element Size** вікна властивостей вузла **Mesh 1**.



Якщо в списку **Sequence type** цієї ПВ вибрати `User-controlled mesh`, то в дереві моделі в вузлі **Mesh 1** з'являться два підвузли **Edge 1** і **Size**. В ПВ останнього можна явно задати довжину скінченного елемента.



В списку **Sequence type** залишимо значення `Physics-controlled mesh`, а в списку **Element Size** виберемо значення `Normal`.

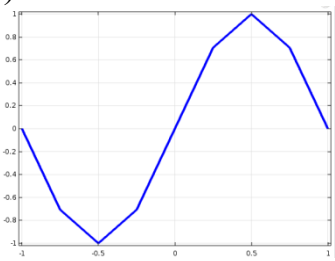
Оскільки вузла **Study** в нашій моделі немає, то виконувати обчислення за допомогою команди Compute непотрібно, і можна відразу перейти до роботи з гілкою результатів **Results**.

В гілку даних **Data Sets** дерева моделі додамо вузол, який буде представляти джерело даних, що складається з нашої скінченно-елементної сітки. Клацніть правою кнопкою по вузлу **Data Sets** і в контекстному меню оберіть Mesh. В ПВ вузла **Mesh 1**, що з'явився в дереві моделі, в списку **Mesh** виберіть `Mesh 1`. Якби в нашій моделі було створено кілька сіток скінченних

елементів, то можна було б вибрати будь-яку з них. Перейменуйте вузол **Mesh 1**, надрукувавши в полі **Label** текст `DataMesh`.

Тепер в гілку **Results** додайте графічну групу **1D Plot Group** з вузлом типу **Line Graph**. У списку **Data set** ПВ вузла **1D Plot Group 1** виберіть ім'я `DataMesh`, а в списку **Data set** ПВ вузла **Line Graph 1** виберіть `From parent`. В ПВ вузла **Line Graph 1** в списку **Selection** виберіть `All domains` або, як альтернативу, виберіть `Manual` і додайте індикатор відрізка, клацнувши в графічному вікні по відрізку лівою кнопкою миші.

Далі чинимо так само, як при побудові одновимірних графіків в прикладі 1. У вікні властивостей вузла **Line Graph 1** в розділі **`y-Axis Data`** в полі **Expression** вводимо вираз, наприклад `sin(pi*x)`. В розділі **`x-Axis Data`** в списку **Parameter** вибираємо `Expression`, і в поле **Expression** вводимо `x`. В розділі **`Coloring and Style`** задаємо стиль, колір і товщину графіка, і натискаємо на кнопку Plot. В графічному вікні з'являється графік функції, побудований по вузлах скінченно-елементної сітки. Щоб продемонструвати те, що графік будується по вузлах, ми використали «грубу» сітку (у вас графік повинен вийти більш гладким).



Побудова графіків поверхонь виконується аналогічно:

- створюється модель розмірності 2D;
- у вузлі **Geometry** створюється двовимірна область;
- в області будується скінченно-елементна (СкЕ) сітка;
- в гілці **Data Sets** створюється **Mesh** вузол, пов'язаний з СкЕ сіткою;
- створюється група двовимірних графіків, які використовують вузол **Data Sets–Mesh** як джерело даних;
- додається вузол конкретного графічного типу, в поле **Expression** якого вводиться бажаний вираз.

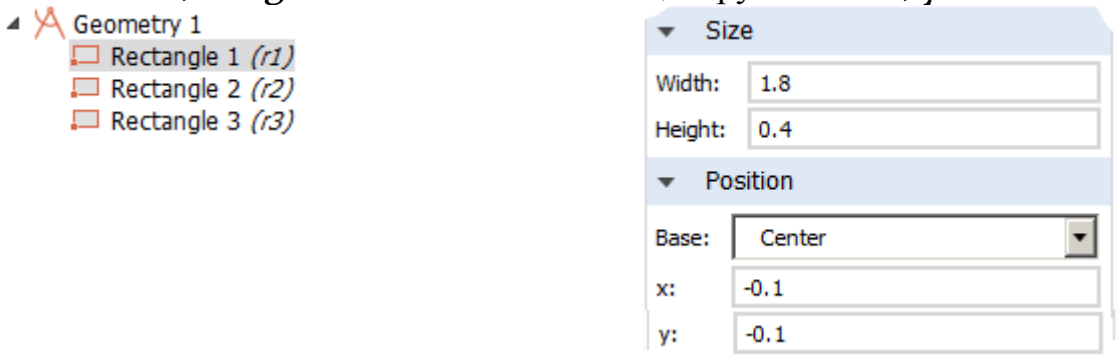
Замість кроків 2 і 3 (створення двовимірної області і побудови в ній СкЕ сітки) інколи використовується двовимірне джерело даних **Grid 2D**.

Приклад 4. Побудуємо графік функції двох змінних над областю у вигляді «скоби» .

Створіть порожню модель. Виберіть пункт меню File->New і у відкрившомуся вікні виберіть створення `Blank Model` моделі.

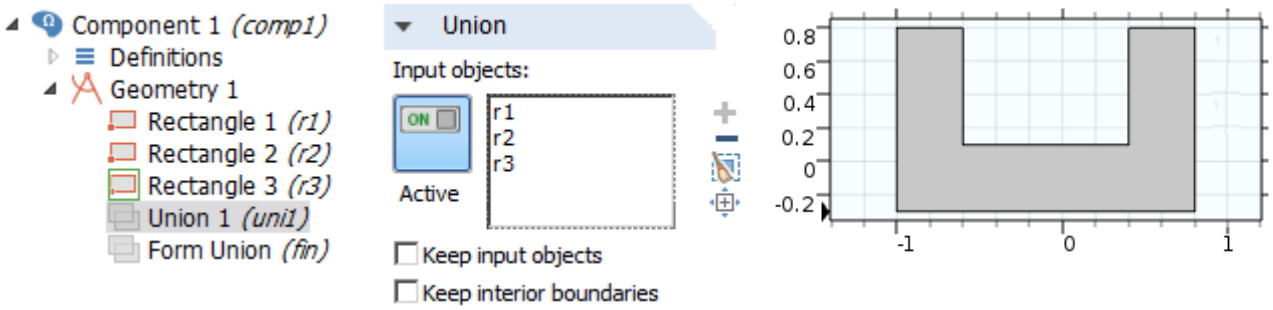
Клацніть правою кнопкою миші по найвищому вузлу дерева моделі (**Untitled.mph**) і в випадному меню виберіть команду Add Component->2D. З'явиться вузол **Component 1** і кілька його підвузлів, з яких нас цікавлять вузли **Geometry 1** і **Mesh 1**. Виділіть вузол **Geometry 1**, на стрічці панелей відкрийте закладку **Geometry** і на ній клацніть по кнопці Rectangle. У випадному списку

виберіть Rectangle (Center) і в графічній області мишею намалюйте прямокутник приблизно такого положення і розміру, як ви бажаєте. В ПВ вузла **Rectangle 1** підправте (якщо треба) параметри прямокутника: задайте розміри **Width** = `1.8`, **Height** = `0.4` і положення центру **x** = `-0.1`, **y** = `0.1`.



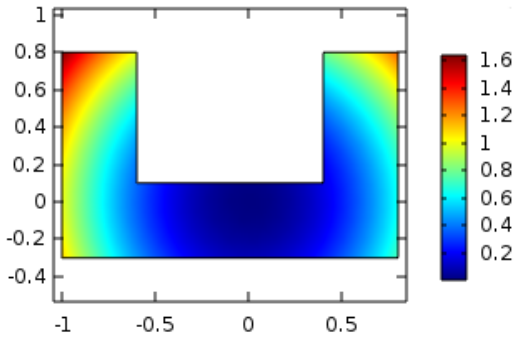
Аналогічно додайте ще два прямокутника. В ПВ вузла **Rectangle 2** введіть **Width** = `0.4`, **Height** = `0.8`, **x** = `-0.8`, **y** = `0.4`. В ПВ вузла **Rectangle 3** введіть **Width** = `0.4`, **Height** = `0.8`, **x** = `0.6`, **y** = `0.4`.

На наступному кроці об'єднайте прямокутники в єдину область, додавши вузол типу **Union**. Для цього клацніть правою кнопкою миші по вузлу **Geometry 1** і в контекстному меню виберіть команду Booleans and Partitions–Union. З'явиться вузол **Union 1**. Активізуйте його і в графічній області по черзі клацніть мишею по кожному з трьох прямокутників. Зніміть перемикач **Keep interior boundaries** і вгорі ПВ натисніть кнопку Build All Objects.

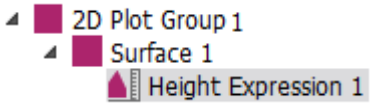


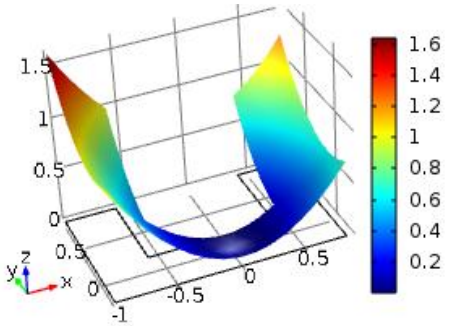
В графічній зоні з'явиться зображення області в формі скоби. На закладці `Mesh` натисніть кнопку Build Mesh, і в області з'явиться скінченно-елементна сітка. В гілку даних **Data Sets** дерева моделі додайте джерело даних, яке буде представляти сітку. Для цього клацніть правою кнопкою миші по вузлу **Data Sets** і в контекстному меню оберіть Mesh. В ПВ вузла **Mesh 1** в списку **Mesh** виберіть `Mesh 1`.

В гілку **Results** додайте графічну групу **2D Plot Group** з підвузлом типу **Surface**. В списку **Data set** ПВ гілки **2D Plot Group 1** виберіть `Mesh 1`, а в списку **Data set** ПВ вузла **Surface 1** виберіть `From parent`. Далі, в ПВ вузла **Surface 1** в розділі **`Expression`** в поле **Expression** введіть вираз, графік якого бажаєте побудувати, наприклад, `x^2+y^2`. Натисніть на кнопку Plot і отримайте наступне зображення.



Щоб побачити графік поверхні в вузол **Surface 1** додайте підвузол типу **Height Expression**.





■

Інколи в полях **Expression** потрібно використовувати «кускові» вирази. В Comsol Multiphysics їх можна створювати в стилі Matlab. Якщо логічний вираз з'являється в аналітичній формулі, то він інтерпретується як +1, якщо умова істинна, або як 0 – якщо умова хибна. Наприклад, вираз $\begin{cases} 0 \ , x \le 0 \\ x^2, x > 0 \end{cases}$ можна записати у вигляді $(x > 0) * x^2$. Також це можна записати як $\text{if}(x > 0, x^2, 0)$. Якщо логічний вираз в першому аргументі оператора `if(...)` істинний, то обчислюється вираз в другому аргументі, інакше обчислюється третій аргумент.

Приклад 5. Побудуємо графік поверхні півсфери, яка задана виразом

$$z = \begin{cases} \sqrt{1 - x^2 - y^2} \ , x^2 + y^2 < 1 \\ \quad 0 \qquad , x^2 + y^2 \ge 1 \end{cases}.$$

Звичайне використання формули $z = \sqrt{1 - x^2 - y^2}$ буде приводити до помилки, оскільки для деяких (x, y) підкореневий вираз буде від'ємним. В COMSOL Multiphysics наш вираз можна записати у вигляді

```
sqrt((x^2+y^2<1)*(1-x^2-y^2)),
```

або у вигляді

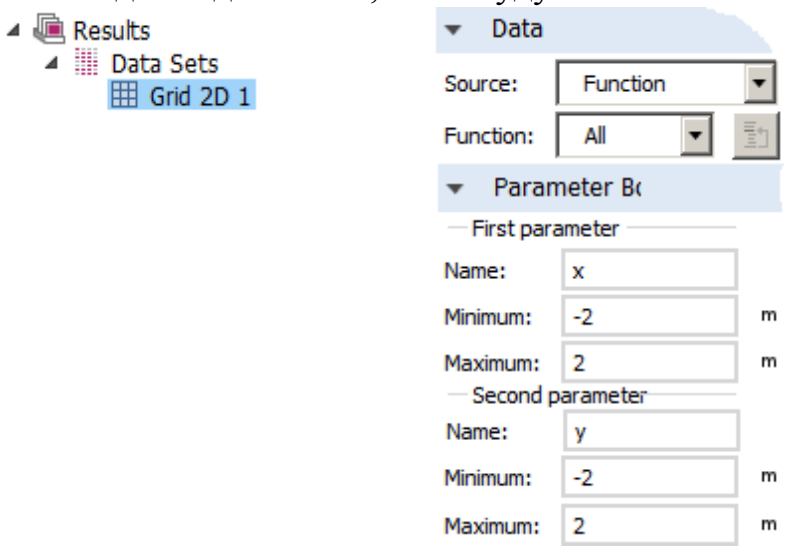
```
if(x^2+y^2<1,sqrt(1-x^2-y^2),0).
```

Створіть порожню модель. Виберіть пункт меню File->New і у відкрившомуся вікні виберіть створення порожньої моделі (`Blank Model`).

Створіть глобальну функцію $f(x, y) = 1$. Це функція заглушка, вона не буде використовуватися, але для нормального функціювання джерел даних `Grid` в файлі моделі має бути присутня принаймні одна глобальна функція.

Клацніть правою кнопкою миші по вузлу дерева моделі **Global Definitions** і виберіть команду Functions->Analytic. В ПВ створеного вузла в поле **Function name** введіть ім'я функції, наприклад, `f`. В полі **Arguments** через кому перелічіть імена аргументів функції `x`, `y`. В поле **Expression** введіть 1.

В гілку **Results - Data Sets** додайте джерело даних **Grid 2D 1**. Для цього клацніть правою кнопкою миші по вузлу **Data Sets** і в контекстному меню виберіть команду More Data Sets - Grid 2D. В ПВ з'явившегося вузла **Grid 2D 1** в якості джерела даних в списку **Source** оберіть `Function`, в списку **Function** виберіть `All`. Потім задайте діапазони, яким будуть належати змінні `x` і `y`.

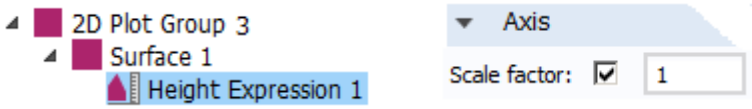


В гілку **Results** додайте вузол **2D Plot Group 1** з підвузлом **Surface 1**. У списку **Data set** ПВ вузла **2D Plot Group 1** оберіть `Grid 2D 1`, а в списку **Data set** ПВ вузла **Surface 1** виберіть `From parent`. В ПВ вузла **Surface 1** в розділі **`Expression`** в поле **Expression** введіть вираз

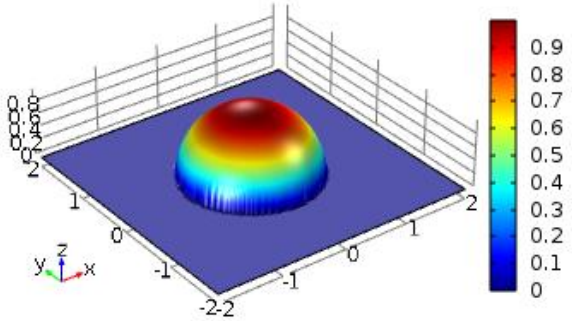
```
sqrt((x^2+y^2<1)*(1-x^2-y^2)),
```

або вираз `if(x^2+y^2<1,sqrt(1-x^2-y^2),0))`.

Вузлу **Surface 1** додайте підвузол типу **Height Expression**, в ПВ якого в поле **Scale factor** введіть одиницю, попередньо встановивши відповідний прапорець.



Натисніть кнопку Plot і отримайте наступне зображення.



▪

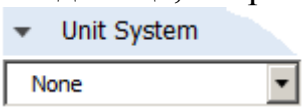
Особливість системи Comsol Multiphysics полягає в тому, що графічні вузли, які використовуються при побудові графіків, повинні бути прив'язані до певного

джерела даних (`Data set`). Їх можна створити штучно так, як ми це робили в попередніх прикладах, або розв'язати яку-небудь задачу і її розв'язок використовувати в якості джерела даних для графіків.

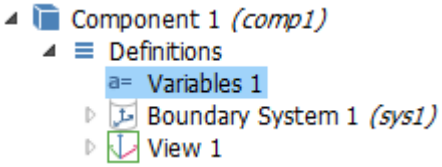
Приклад 6. Побудуємо графік тривимірної кривої, наприклад, гвинтової лінії (спіралі) з рівнянням $x = \cos t, y = \sin t, z = t/8$. Для цього створимо тривимірну динамічну модель зі змінним `u,v,w`, які будуть залежати від змінної часу `t` по зазначеним формулам. Прив'язуючись до цих змінних, ми побудуємо криву.

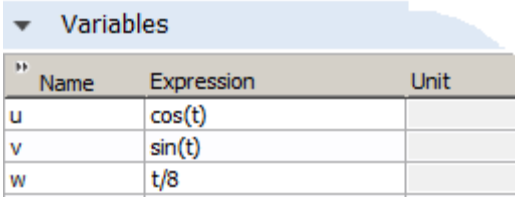
Виберіть пункт меню File->New і у відкрившомуся вікні виберіть створення порожньої моделі (`Blank Model`).

В ПВ кореневого вузла **Untitled.mph (root)** в списку **`Unit system`** відключіть використання систем одиниць, вибравши зі списку значення `None`.

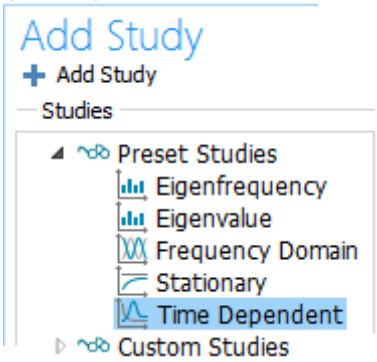


Клацніть правою кнопкою миші по найвищому вузлу дерева моделі (**Untitled.mph (root)**) і в випадному меню виберіть команду Add Component-> 3D. З'явиться вузол **Component 1** і кілька його підвузлів, з яких нас цікавить вузол **Definitions**. Клацніть по ньому правою кнопкою миші і з випадного меню виберіть команду Variables. В ПВ створеного вузла **Variables 1** знаходиться таблиця, призначена для задання змінних. Заповніть її так, як показано нижче.



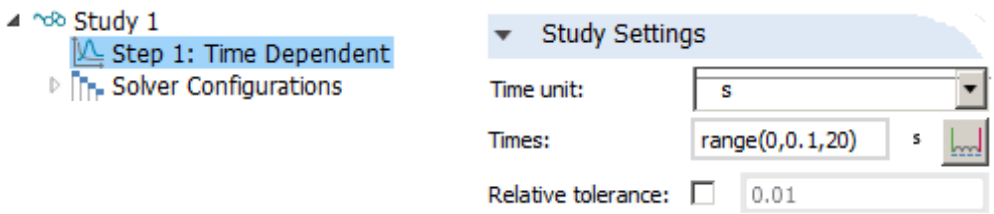


Клацніть правою кнопкою миші по кореневому (найвищому) вузлу дерева моделі (**Untitled.mph (root)**) і в випадному меню виберіть команду Add Study. Система відкриє вікно вибору типу обчислення (`Add Study`).

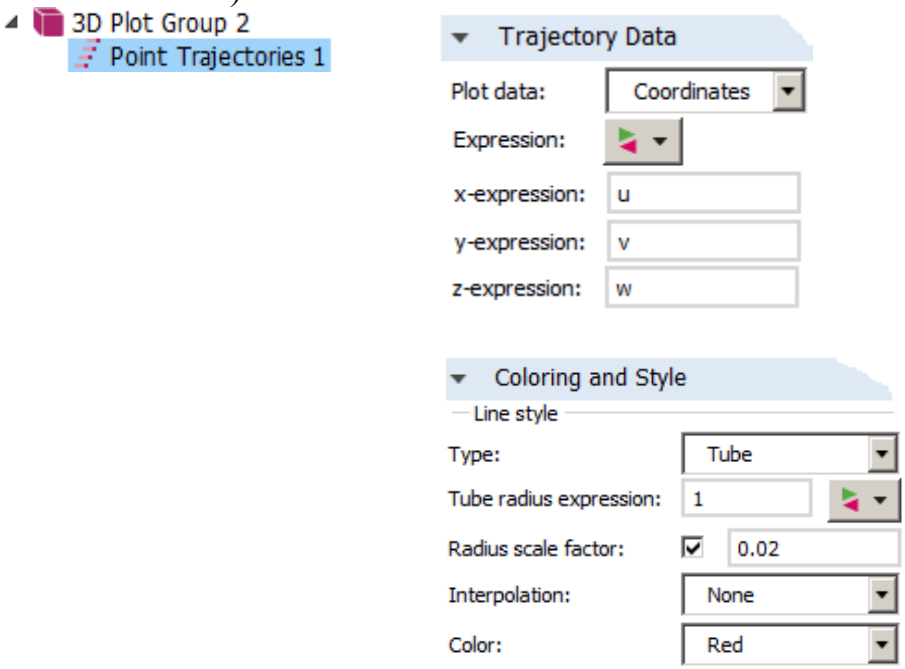


Виконайте подвійне клацання по рядку `Time Dependent`, який означає, що змінні моделі можуть залежати від часу. Закрийте вікно хрестиком в правому верхньому куті. В дереві моделі з'явиться вузол **Study 1**.

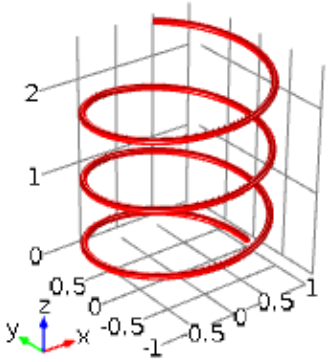
В ПВ вузла **Study 1 - Step 1: Time Dependent** в полі **Times** задайте діапазон зміни параметра `t`. Функція `range(0,0.1,20)` представляє значення параметра `t`, що змінюються від `0` до `20` з кроком `0.1`. Нижче в полі **Relative Tolerance** можна задати відносну похибку.



На закладці Home натисніть кнопку = Compute . Обчислення моделі буде виконано і в гілці **Results - Data Sets** з'явиться вузол джерела даних **Study 1/Solution 1.** Щоб його використати для побудови кривої, в гілку **Results** додайте вузол тривимірної графічної групи. Для цього клацніть правою кнопкою миші по вузлу **Results** і в випадному меню оберіть команду 3D Plot Group. Зверніть увагу, що в ПВ створеного вузла **3D Plot Group 1** в списку **Data set** автоматично обрано Study 1/Solution 1, тобто нова графічна група прив'язана до джерела даних, що збігається з розв'язком моделі. Перевірте, що в полі **Time(s)** стоїть число 20 – максимальний момент часу із заданого нами діапазону. Клацнувши по вузлу **3D Plot Group 1** правою кнопкою миші, виберіть команду More Plots - Point Trajectories. З'явиться підвузол **Point Trajectories 1**. В його ПВ в поля **x-expression**, **y-expression**, **z-expression** введіть u,v,w. Також в цій ПВ в розділі **Coloring and Style** задайте параметри зображення кривої так, як показано нижче (показані тільки деякі розділи вікна властивостей).



Клацання по кнопці Plot вгорі ПВ вузла **Point Trajectories 1** будує криву.



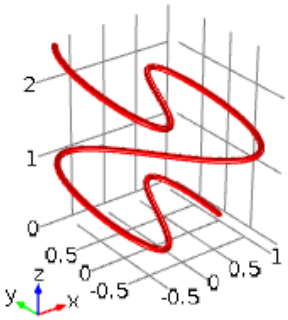
Для побудови кривих необов'язково створювати змінні моделі. Можна безпосередньо в полях виразів графічних вузлів вводити формули. Наприклад, додамо ще один вузол **3D Plot Group 2**. В його ПВ в списку **Data Set** знову виберіть `Study 1/Solution 1`. Додайте в **3D Plot Group 2** підвузол **Point Trajectories 1**. В його ПВ в розділі **`Trajectory Data`** введіть такі вирази

x-expression: cos(t/2)

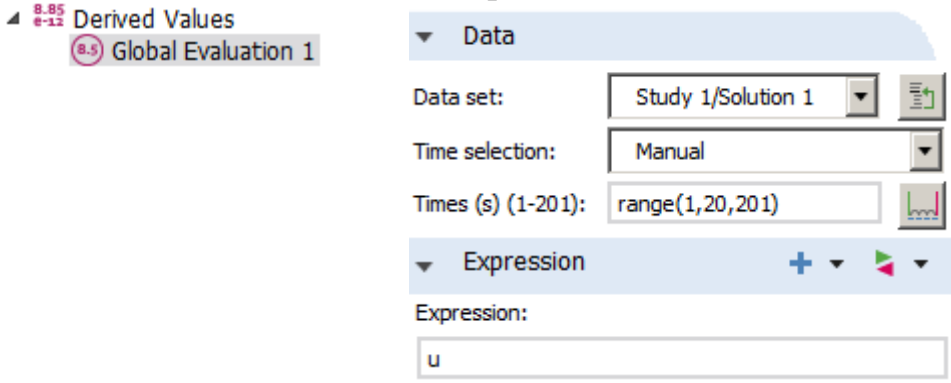
y-expression: sin(t)

z-expression: t/8

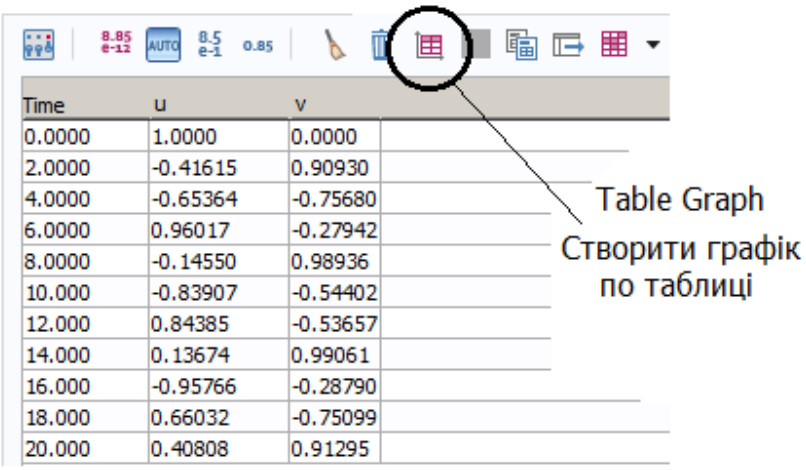
Клацання по кнопці Plot вгорі ПВ вузла **Point Trajectories 1** побудує просторову криву, не пов'язану зі змінними `u`, `v`, `w`. Прив'язка існує тільки до діапазону зміни параметра `t`, який заданий нами раніше у вигляді `range(0,0.1,20)`.



Припустимо тепер, що ви бажаєте надрукувати таблицю значень функцій `u` і `v`. Для цього призначені підвузли гілки **Derived Values**. Клацніть по вузлу **Derived Values** правою кнопкою миші і в випадному меню оберіть Global Evaluation. В ПВ створеного вузла **Derived Values - Global Evaluation 1** в писку **Time selection** виберіть `Manual`, і в полі **Times (s) (1-201)** задайте список номерів моментів часу (не самі моменти), які бажаєте використати, наприклад, `range(1,20,201)`. Це означає, що вибирається кожен двадцятий момент часу.

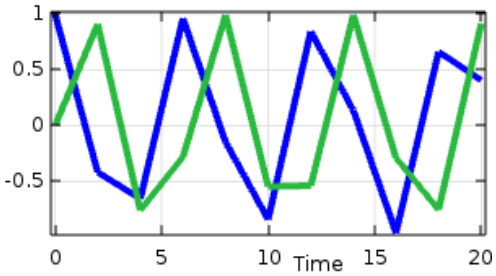


Кнопкою Evaluate, розташованою вгорі ПВ, побудуйте таблицю, яка містить два стовпці: моменти часу і значення функції `u`. Щоб додати в таблицю значення другої змінної, потрібно створити схожий вузол для змінної `v`. Клацніть по вузлу **Derived Values** правою кнопкою миші і в випадному меню оберіть Global Evaluation. В ПВ створеного підвузла **Derived Values - Global Evaluation 2** в полі **Expression** введіть `v`, і в кнопці-списку Evaluate оберіть ім'я вже створеної таблиці **Table 1 - Global Evaluation 1**. Діапазон моментів часу можна не задавати (він буде взятий з таблиці).



Time	u	v
0.0000	1.0000	0.0000
2.0000	-0.41615	0.90930
4.0000	-0.65364	-0.75680
6.0000	0.96017	-0.27942
8.0000	-0.14550	0.98936
10.000	-0.83907	-0.54402
12.000	0.84385	-0.53657
14.000	0.13674	0.99061
16.000	-0.95766	-0.28790
18.000	0.66032	-0.75099
20.000	0.40808	0.91295

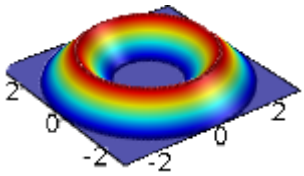
По табличним значенням можна будувати графіки. Кнопка Table Graph, яка розташована над таблицею, створить одновимірну графічну групу і побудує графіки кривих за значеннями в стовпцях цієї таблиці. При цьому перший стовпець буде використаний в якості абсцис вершин ламаних.



Зауваження. Не називайте змінні та шукані функції іменами, що збігаються з назвами просторових координат. Наприклад, в поточному прикладі в таблиці змінних **Variables** не варто використовувати літери `x`, `y`, `z` в якості імен змінних (ми застосували імена `u,v,w`). Інакше деякі з побудов не працюватимуть. Повідомлень про помилки може не бути, але графіки немалюватимуться.

Вправа 1. Використовуючи логічні операції в арифметичному виразі, побудуйте в області $-3 \le x \le 3$, $-3 \le y \le 3$ графік функції

$$f(x,y)=\begin{cases} 0 \quad , & x^2+y^2 \ge 9 \\ 1-\left|\sqrt{x^2+y^2}-2\right| \ , & 1<x^2+y^2<9 \\ 0 \quad , & x^2+y^2<1 \end{cases}$$



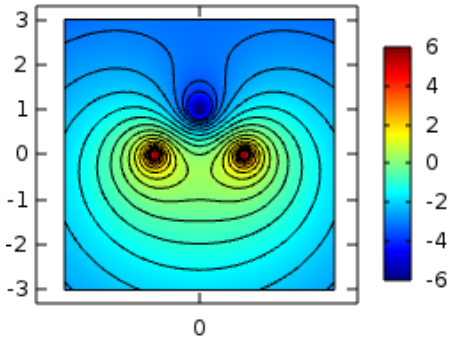
В математичних виразах Comsol функція $\sqrt{x}$ позначається як `sqrt(x)`, а $|x|$ як `abs(x)`. Логічні операції **І** та **АБО** позначаються `&&` та `||`. Наприклад, подвійна нерівність $1<x<2$ записується як `1<x && x<2`.

Вправа 2. Графічно зобразити двовимірне електростатичне поле.

Електростатичний потенціал прямої перпендикулярної до площині `z=0`, яка проходить через точку (x_0, y_0), обчислюється за формулою

$$\varphi(\mathbf{r},\mathbf{p}_0)=2q_0 \ln \frac{1}{\sqrt{(x-x_0)^2+(y-y_0)^2}}.$$

де q_0 – лінійна густина заряду уздовж прямої. Потенціал поля n прямих внаслідок суперпозиції силових полів визначається формулою $\varphi = \sum_{i=1}^{n} \varphi(\mathbf{r}, \mathbf{p_i})$. За допомогою двовимірного суцільного та лінійного контурного графіків побудувати поле трьох точкових зарядів $q_1 = 1, q_2 = 1, q_3 = -1$, розташованих в точках $(1,0), (-1,0), (0,1)$.



Для виразу $\varphi(\mathbf{r}, \mathbf{p}_0)$ створіть функцію. Натуральний логарифм $\ln x$ в Comsol позначається як `log(x)`.

Щоб отримати наведений вище рисунок, при побудові суцільного контурного графіка обмежте зверху діапазон значень кольорів, а для контурного графіка підберіть значення ліній рівня потенціалу.

1.4. Термінологія, позначення і скорочення.

В цьому параграфі описуються домовленості, які будуть застосовуватися на протязі всього посібника для скороченого запису команд та інструкцій.

В посібнику часто використовуються наступні скорочення:

ПВ , ВВ	панель властивостей, вікно властивостей
ДМ	дерево моделей
СкЕ	скінченні елементи, скінченно-елементний
МСЕ, МСкЕ	метод скінченних елементів
ДР	диференціальне рівняння
ЗДР	звичайне диференціальне рівняння
ДРЧП	диференціальне рівняння в частинних похідних
ГВ	графічне вікно
ДД, НД	джерело даних, набір даних (Data Sets)

В COMSOL Multiphysics команди можуть виконуватися вибором відповідного елемента з головної панелі інструментів (стрічки інструментальних панелей), з випадного меню, що відкривається після клацання правою RMB (Right Mouse Button) кнопкою миші по вузлу ДМ, або клацанням лівою LMB (Left Mouse Button) кнопкою миші по кнопці в ПВ.

При запису назви вузла часто будемо вказувати ім'я батьківської гілки, наприклад, **Data Sets** (якщо потрібно), а потім через тире - ім'я вузла **Grid 1D 1**. Для цих імен буде використовуватися шрифт **Times New Roman жирний**. Таким чином, запис **Data Sets - Grid 1D 1** репрезентує ім'я вузла **Grid 1D 1**, який розташований в гілці **Data Sets** дерева моделі.

Для запису команд ми будемо використовувати різні скорочені позначення. Значок **>** (або **->**) відокремлює ім'я однієї команди від імені іншої. Назва активного вузла ДМ може стояти перед командою і відділятися від неї двокрапкою. Наприклад, запис

Study 1 : RMB > Update Solution

означає, що слід виконати клацання правою кнопкою миші (RMB) по вузлу **Study 1** і в випадному меню вибрати команду Update Solution.

Часто назва активного вузла і команди розносяться на різні рядки, наприклад,

Study 1 :

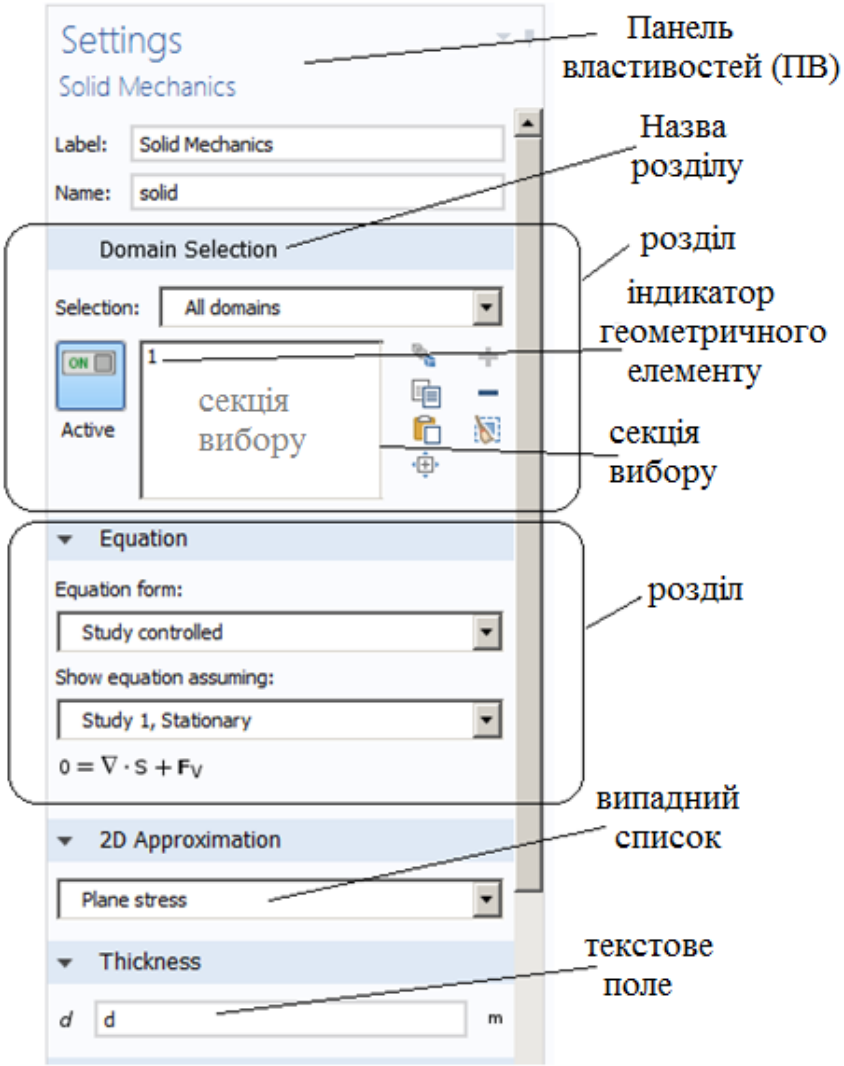
RMB > Update Solution

Запис LMB > Plot без імені вузла означає клацання лівою (LMB) кнопкою миші по кнопці Plot в ПВ активного вузла. Наприклад, якщо активним є вузол **1D Plot Group 1 - Line Graph 1**, то останній запис є скороченням інструкції

1D Plot Group 1 – Line Graph 1: LMB > Plot

В тексті посібника ім'я вузла і назва команди набираються різними шрифтами. Якщо з контексту ясно, де вузол розташований, то назва батьківського вузла не вказується.

На наступному рисунку наведено пояснення деяких термінів, які використовуються в посібнику.



Основний потік дій в ході моделювання задається порядком наступності команд. Команди, які йдуть поспіль і не потребують аргументів, будемо записувати коротко. Наприклад, послідовність команд при створенні нової моделі може бути представлена рядком

File->New->Blank Model

або, наприклад, довшим рядком

File -> New -> Model Wizard -> 2D -> Mathematics – Classical PDEs – Poisson's Equation(poeq) ->Stationary

Тут ми перелічуємо елементи інтерфейсу, які вибираються на кожному послідовному етапі роботи конструктора (`Model Wizard`) при створенні моделі задачі. Стрілки **->** відокремлюють назви послідовно вибираних елементів інтерфейсу (фактично команд), а тире розділяє назви вузлів і підвузлів дерева, в якому виконується вибір елемента.

Послідовність команд, які використовуються при створенні нового елементу моделі (тобто нового вузла ДМ), будемо записувати наступним рядком:

`Ім'я панелі ->` **Назва групи команд ->** Команда

При цьому імена панелей, груп команд (або кнопок-списків) і самі команди будемо розділяти стрілками, і виділяти різним шрифтом. Наприклад, команда

`Results->`More Data Sets - Grid 1D.

означає, що на панелі `Results` слід розкрити кнопку-список **More Data Sets** і в випадному списку вибрати команду Grid 1D.

Пам'ятайте, що доступ до команд можна отримати різними способами. Один з них полягає у використанні головної інструментальної панелі (стрічки панелей). Інший полягає у виборі команд з контекстного меню, яке відкривається при натисканні правою кнопкою миші по відповідному вузлу дерева моделі. Ми завжди будемо описувати тільки один зі способів, але ви можете використовувати будь-який, зручний для вас.

Наші дії при заповненні панелей властивостей вузлів теж будемо записувати коротко. Наприклад, введення параметрів в ПВ вузла **Data Sets - Grid 1D 1** будемо записувати в такий спосіб

Data Sets – Grid 1D 1:

Data – Source = `Function`

Data – Function = `All`

Parameter Bounds – Name = `x`

Parameter Bounds – Minimum = `-1`

Тут перший рядок репрезентує ім'я активного вузла дерева моделі. Кожний наступний рядок представляє в ПВ цього вузла ім'я заповнюваного поля і значення, що вводиться або вибирається зі списку. Зліва від знака рівності стоїть ім'я поля (або списку), яке складається з імені розділу, в якому це поле знаходиться, і самого імені поля. Ім'я розділу відділяється від імені поля знаком тире. Для назви розділу буде використовуватися шрифт **Courier New жирний**, а для імені поля - шрифт **Constantia жирний**. Для значень, які стоять праворуч від знака рівності і вводяться або вибираються зі списку, буде

використовуватися шрифт `Courier New`. Рядки команд для полів, вміст яких не змінюється (використовуються значення за замовчуванням), ми записувати не будемо.

Щоб багаторазово не друкувати назву повторюваних розділів в кожному рядку, ці назви ми інколи будемо опускати. Наприклад, наведені вище інструкції ми іноді будемо записувати в такий спосіб

Data Sets – Grid 1D 1:

`Data` – **Source** = `Function`

– **Function** = `All`

`Parameter Bounds` – **Name** = `x`

– **Minimum** = `-1`

Коли налаштування задаются за допомогою перемикачів або радіокнопок, то щоб вказати значення таких елементів ми будемо присвоювати їх іменам логічні константи `True` або `False`. Наприклад,

`Scale` – **Scale factor** = `True` (*включити*)

Часом іменам таких елементів будемо просто присвоювати слова `Включити` та `Виключити`.

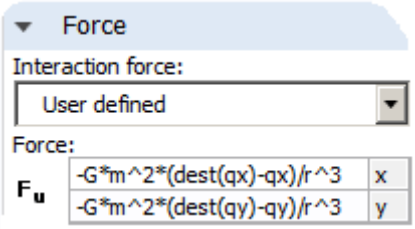
`Heat Source` – **General source** = `Виключити` (*виключити радіокнопку*)

Інколи в розділах зустрічаються поля без міток. Для їх позначення будемо використовувати кутові дужки **< >**. Наприклад, запис

Untitled.mph (root): `Unit System` – **<>** = `None`

означає, що в ПВ кореневого вузла ДМ в розділі **`Unit System`** в списку без мітки слід вибрати значення `None`.

Іноді мітка відноситься до декількох полів, наприклад, коли вона позначає вектор.



Для розрізнення таких полів ми будемо використовувати мітки в кутових дужках. Наприклад, для скороченого запису вводу даних, показаних на попередньому малюнку, ми будемо використовувати такі рядки

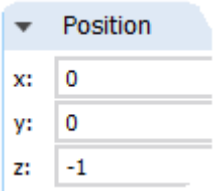
`Force` – **Force – Fu - <x>** = `-G*m^2*(dest(qx)-qx)/r^3`

`Force` – **Force – Fu - <y>** = `-G*m^2*(dest(qy)-qy)/r^3`

Завдання значень в три схожі поля одного розділу будемо записувати одним рядком. Наприклад, запис

`Position` – **x, y, z** = `0,0,-1` (*три поля*)

означає, що треба виконати наступні введення



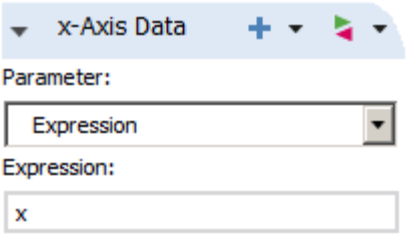
Інструкції введення/вибору значень в різнойменні поруч розташовані поля одного розділу ми також інколи будемо записувати одним рядком. Наприклад, замість двох рядків

`x-Axis Data` – **Parameter** = `Expression`

`x-Axis Data` – **Expression** = `x`

будемо писати один

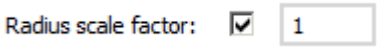
`x-Axis Data` – **Parameter, Expression** = `Expression,x` (*два поля*)



Праворуч від значень ми іноді будемо додавати коментар – текст в дужках. В останній інструкції коментар пояснює факт введення значень в два поля, а не в одне.

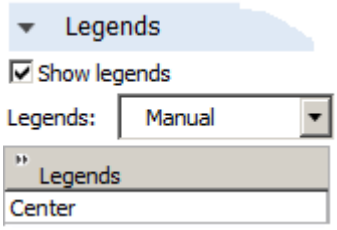
Якщо навпроти поля треба встановити прапорець і ввести значення, то обидві дії будемо записувати одним рядком, наприклад,

`Coloring and Style` – **Radius scale factor** = `True,1`



Якщо в схожій інструкції друге значення повинно бути рядком, то його в нашому посібнику ми будемо записувати в лапках (щоб не виникало непорозуміння), хоча вводити лапки в текстове поле ми не будемо. Наприклад,

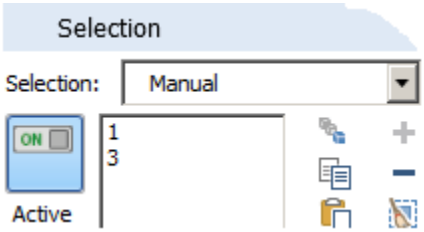
`Legends` – **Show legends, Legends** = `True, 'Center'`



Схожі скорочення будемо використовувати в секціях вибору. Наприклад, інструкція

`Selection` – **Selection** = `Manual;1,3`

позначає, що в розділі **`Selection`** у випадному списку **Selection** обрано значення `Manual`, а нижче в секції вибору – графічні примітиви з номерами/індикаторами `1` і `3`. Тут в якості розділювача ми будемо використовувати крапку з комою та коми.



Для введення значень в таблиці, які зустрічаються в ПВ, ми будемо використовувати схожі позначення. Наприклад, введення значень в перший

рядок таблиці глобальних змінних **Global Definitions - Variables 2** будемо записувати наступними інструкціями

Global Definitions - Variables 2:

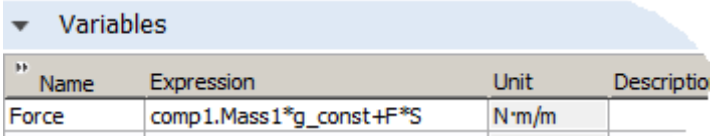
`Variables` – **Name** = `Force`

– **Expression** = `comp1.Mass1*g_const+F*S`

або коротше

`Variables` – **Name, Expression** = `Force, comp1.Mass1*g_const+F*S`

Тут **Name** і **Expression** назви стовпців таблиці, яка знаходиться в розділі **`Variables`**.

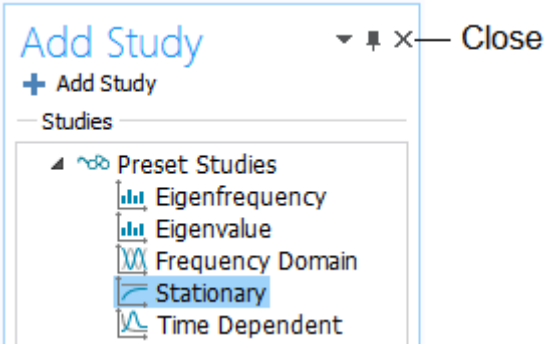


Повторення схожого запису буде означати заповнення наступного рядка таблиці.

Інколи після команди відкривається вікно, в якому треба зробити вибір. Наші дії ми стисло будемо записувати, наприклад, таким рядком:

Untitled.mph (root): RMB > Add Study = `Add Study -> Stationary -> X`

Це буде означати, що після клацання правою кнопкою миші по вузлу **Untitled.mph (root)** і вибору в випадному меню команди Add Study відкриється вікно `Add Study`, в якому треба обрати елемент `Stationary`.



Вибір виконується в стандартний спосіб (в нашому випадку подвійним клацанням миші по рядку `Stationary` або однократним клацанням по кнопці +Add Study вгорі вікна). Потім вікно `Add Study` закривається хрестиком (`X`) в правому верхньому куті.

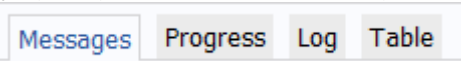
В тих випадках, коли скорочений запис не відповідатиме реальним етапам моделювання, будемо додавати пояснювальний текст.

Поступово ми перейдемо до запропонованого стилю опису процесу створення моделей.

1.5. Початкові відомості про моделювання в COMSOL Multiphysics.

1.5.1. Структура моделей COMSOL Multiphysics.

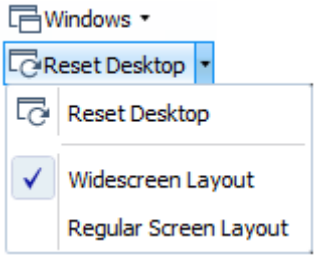
Моделювання в COMSOL Multiphysics починається з вибору розмірності простору і фізичної моделі (стаціонарної або динамічної), тобто зі створення шаблону задачі. Цей шаблон будується в дереві моделі, яке розташовано в панелі конструктора моделей (Model Builder). Спочатку модель має ім'я Untitled, але пізніше (при збереженні в файл) ви надасте їй осмислене ім'я. Зазвичай праворуч від конструктора моделей знаходиться вікно властивостей активного вузла дерева. Ще правіше розташовано графічне вікно, яке використовується для креслення геометрії моделей або для відображення графіків. Нижче графічного вікна розташована зона з ярликами, що представляють вікна повідомлень (Messages), процесу виконання обчислень (Progress), вікно журналу (Log) і вікно таблиць (Table).



Вікно Messages зазвичай відображає повідомлення про помилки. Вікно Progress активно під час обчислень і відображає поточну інформацію про хід виконання завдання. Вікно журналу Log відображає службову інформацію про результати обчислень. При активізації ярлика Table в цій області з'являються таблиці, які зазвичай містять значення розрахункових величин.

Вікно конструктора моделей (Model Builder), вікно властивостей (Settings), графічне вікно, а також вікна Messages, Progress, Log і Table можна розгорнути на все вікно програми. Для цього потрібно виконати подвійне клацання по корінцю вікна. Повторне подвійне клацання по корінцю поверне вікно до початкового розміру.

Протягом роботи ваш екран може приймати різні види. Щоб повернутися до стандартного способу розташування вікон можна на стрічці панелей відкрити закладку Home, розкрити кнопку-список **Reset Desktop** і вибрати один з можливих варіантів: Widescreen Layout (вікна Model Builder, Settings і Graphics розмістяться на екрані зліва направо), Regular Screen Layout (перші два вікна розмістяться одне над одним зліва екрану, а графічне вікно посяде весь простір праворуч від них) або Reset Desktop, щоб відновити колишній вигляд.



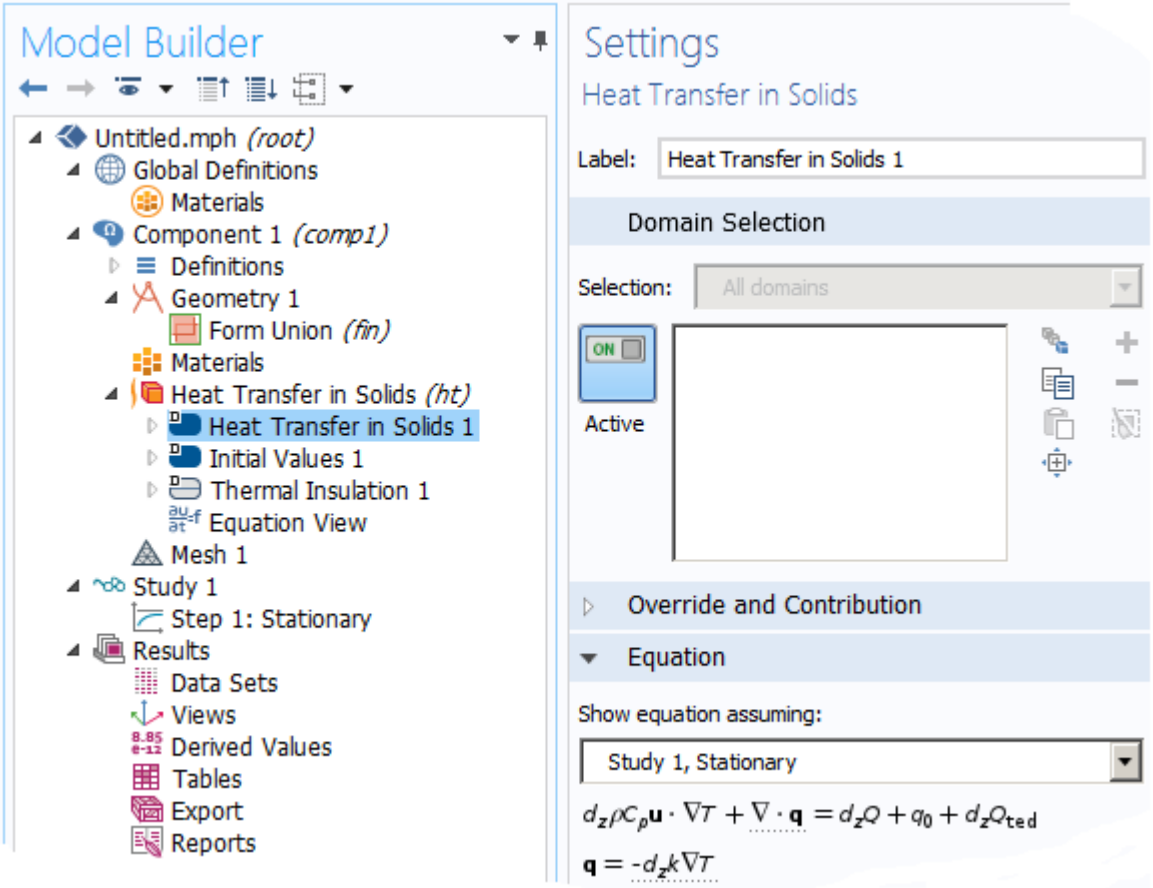
В цьому параграфі ми детально обговоримо послідовність створення та розв’язання двовимірної задачі теплопровідності. Ймовірно, при першому читанні вам не все буде зрозуміло. Познайомтесь з параграфом і поверніться до нього пізніше, коли ми розв’яжемо більше задач, і у вас з'явиться досвід моделювання в COMSOL Multiphysics.

Створіть шаблон моделі двовимірної стаціонарної задачі теплопровідності (heat transfer in solids) наступною послідовністю команд.

File->New->Model Wizard->2D ->

Heat Transfer - Heat Transfer in Solids (ht) ->Stationary

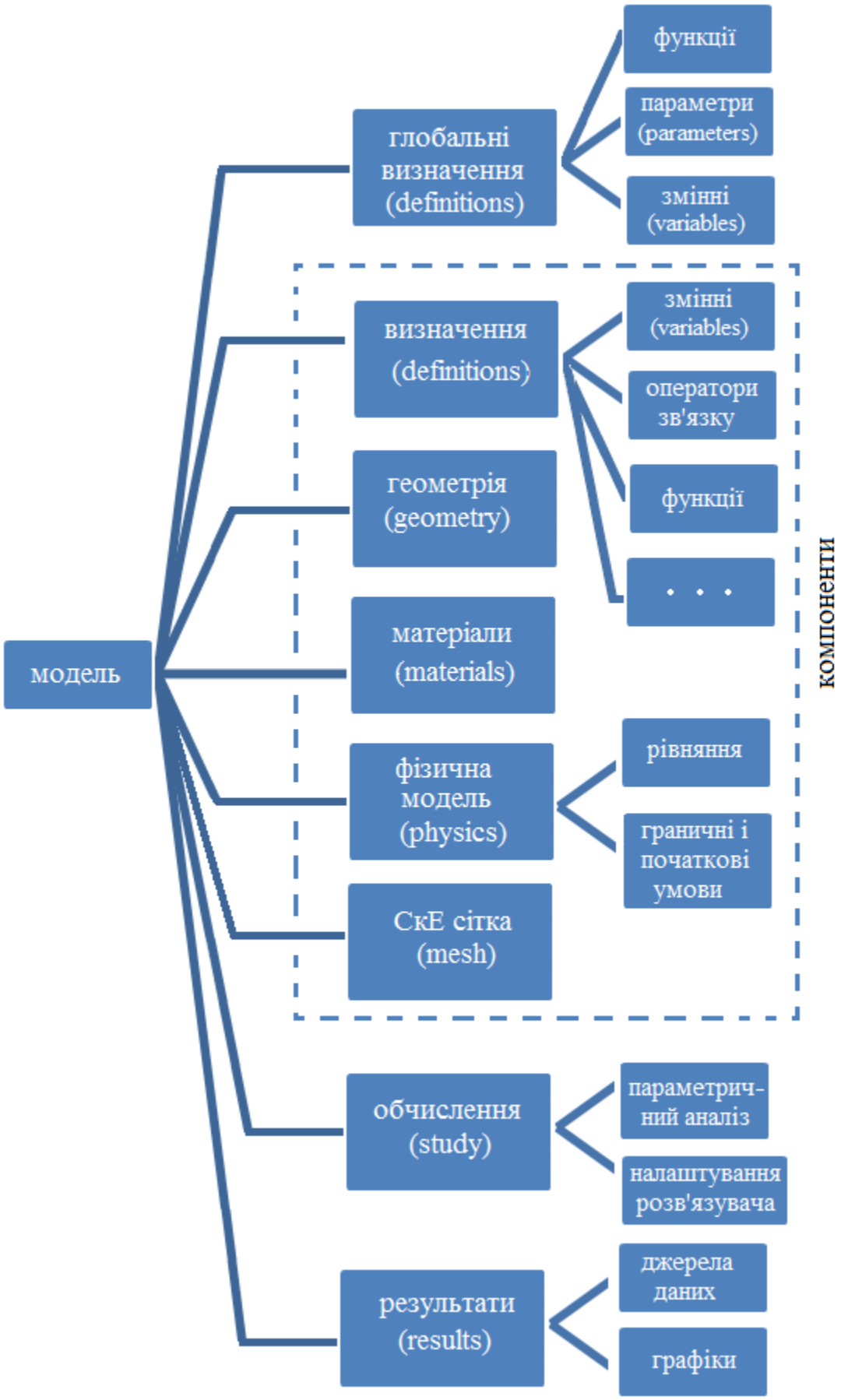
Дерево моделі набуде вигляду, показанному на наступному рисунку зліва.



Праворуч від вікна `Model Builder` буде відкрито вікно `Settings` (**в**ікно **в**ластивостей – ВВ, інакше **п**анель **в**ластивостей – ПВ) активного вузла. Воно призначено для налаштування параметрів відповідного елемента моделі. Елементи дерева моделі можуть мати підвузли, у яких можуть бути свої підвузли і т.д. Основними елементами моделі є гілки: **Global Definitions** (визначення), **Component** (компонент), **Geometry** (геометрія), **Materials** (матеріали), тут **Heat Transfer in Solids** (теплопровідність в твердих тілах), **Mesh** (скінченно-елементна сітка), **Study** (обчислення), **Results** (результати обчислень). Замість **Heat Transfer in Solids** може бути обрана будь-яка фізична модель, що доступна в системі. Ці основні елементи завжди присутні в дереві повноцінної моделі, хоча іноді ми будемо використовувати і неповні моделі. Значення деяких елементів зрозуміло з їх назви, сенс інших ми коротко пояснимо нижче.

Головні елементи будь-якої моделі, створеної в Comsol Multiphysics, і їх взаємозв'язки можна описати наступною схемою.

Гілки **Definitions** (визначення) бувають глобальними (для всієї моделі) і локальними для кожного компонента.



Під компонентом розуміється гілка дерева моделі, яка використовується для об'єднання вузлів пов'язаних фізичних процесів, що відбуваються в спільній просторовій області. Наприклад, в тілі може протікати струм, в наслідок цього

виділятися тепло та відбуватися деформування. Разом ці процеси можна назвати «мультифізикою». Мультифізика – це жаргон. Під цим терміном ми розуміємо будь-яку замкнену систему диференціальних рівнянь з кількома шуканими величинами різної фізичної сутності та розмірності, що розв’язується як одне ціле в спільній просторовій області. Термін «розширена мультифізика» позначає зв'язування рівнянь, які діють в відмінних розрахункових областях. Тоді в моделі присутні кілька компонентів (вузлів **Component**), і для їх зв’язку використовуються глобальні змінні та спеціальні оператори (coupling operators). В нашому посібнику в обох випадках ми будемо використовувати єдиний термін «мультифізика».

Отже колекція вузлів декількох фізичних моделей, що використовують спільну геометричну область, створювану в гілці **Geometry**, в термінології Comsol Multiphysics називається компонентом (component). Окрім гілки **Geometry** та гілок фізичних моделей в цю колекцію включаються вузли **Mesh** та **Materials**. На наведеній схемі елементи одного компонента виділені пунктиром, і таких компонентів може бути декілька.

Гілки **Definitions** зазвичай містять вузли з таблицями параметрів, змінних, а також вузли операторів і функцій. Параметри – це числові константи, імена яких можна використовувати в декількох місцях моделі. Змінні – це назви величин, які можуть залежати від положення спостережуваної точки. Змінними є просторові координати (наприклад, *x*, *y*, *z*) і шукані величини (наприклад, T - температура). В таблиці змінних користувач може створювати власні змінні з інших, використовуючи припустимі операції (наприклад, диференціювання). Оператори – це математичні конструкції, які зазвичай обробляють величини (шукані функції), досліджувані в моделі. Вузли функцій містять «алгоритми» їх обчислення (аналітичні вирази, таблиці для завдання кускових функцій і т.д.). Кожній функції надається ім'я, і в будь-якому виразі, в якому вона потрібна, ви будете використовувати це ім'я.

Елементи глобальної (зовнішньої) гілки визначень **Global Definitions** можна використовувати в будь-яких місцях моделі. Елементи гілки **Definitions**, визначеній в компоненті, призначені для використання в його межах, але, в разі потреби, можуть використовуватися поза компонентом за допомогою точкової нотації.

Гілка **Geometry** містить інструкції креслення розрахункової області компонента. Вони представляються вузлами ДМ. Безпосередньо в ПВ вузла **Geometry** задаються одиниці вимірювання довжин та кутів.

Використовуючи скорочені позначення, описані раніше, запишемо послідовність команд створення геометрії моделі.

`Geometry` **`-> Primitives`** – Rectangle

Component 1 - Geometry 1 - Rectangle1:

`Size` – **Width** = `3`

`Size` – **Height** =`1`

`Position` – **x, y** = `0,0` (*два поля*)

`Geometry` -> **`Primitives`** – Circle

Component 1 - Geometry 1 - Circle 1:

Size and Shape – **Radius:** = 0.25

Position – **x, y** = 1,0.5 (*два поля*)

Geometry -> **Primitives** – Circle

Component 1 - Geometry 1 - Circle 2:

Size and Shape – **Radius:** = 0.25

Position – **x, y** = 2,0.5 (*два поля*)

За допомогою операції логічного віднімання видалимо з області прямокутника область правого кола.

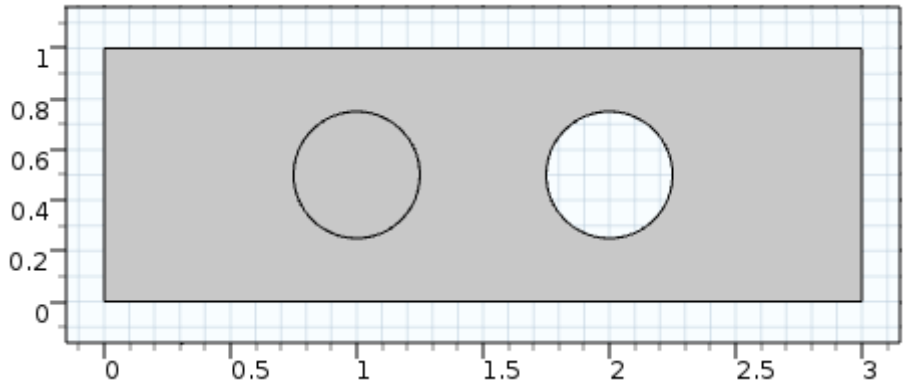
Geometry **->Booleans and Partitions** – Difference

Component 1 - Geometry 1 - Difference 1:

Difference – **Object to add** = r1 (*прямокутник*)

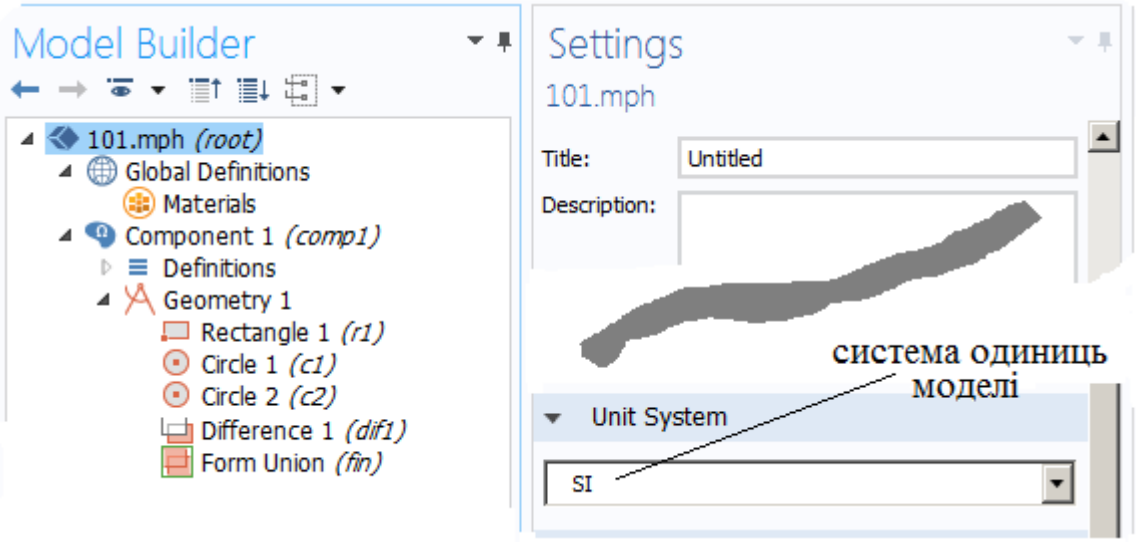
Union – **Object to subtract** = c2 (*правий круг*)

LMB > Build All Objects

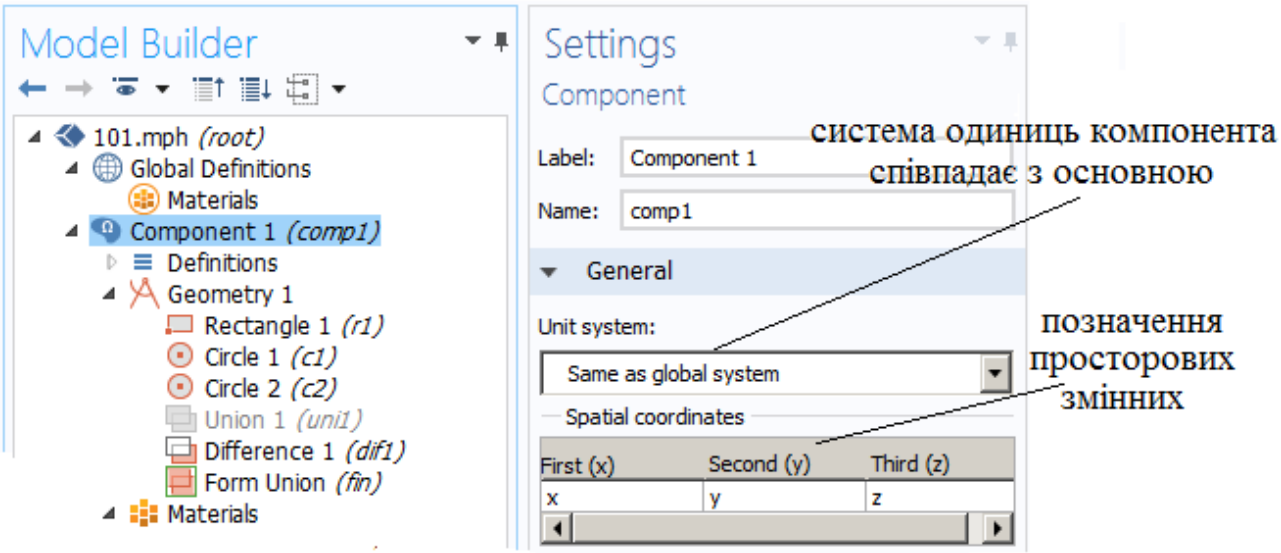


Гілка **Materials** дозволяє вибирати речовини з вбудованою бази, щоб використовувати їх фізичні параметри, а також дозволяє «створювати» власні матеріали.

Всі значення параметрів бажано задавати в системі одиниць, яка використовується в моделі. Дізнатися (вибрати) основну систему одиниць можна в ПВ вузла самого верхнього рівня (**root**) в розділі **Unit System**.



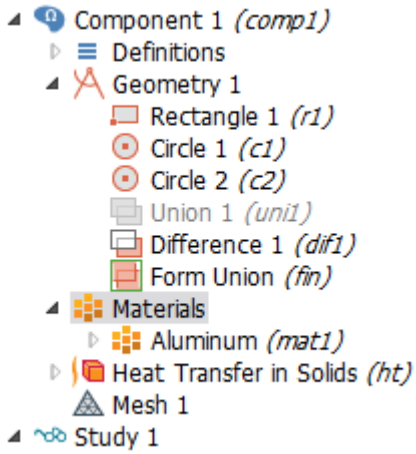
Кожен компонент може мати власну систему одиниць, назву якої можна дізнатися і змінити в ПВ вузла **Component**.

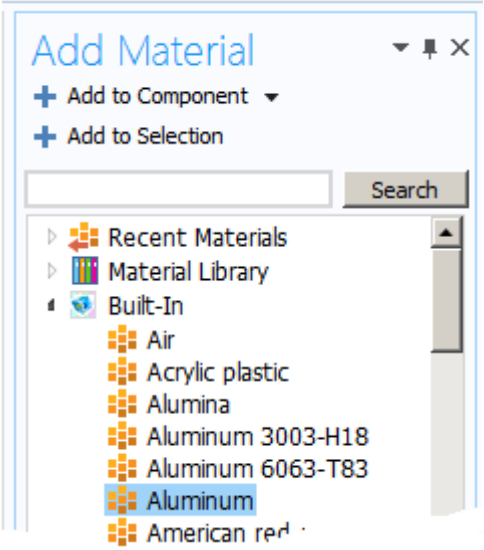


Створіть в моделі вузол матеріалу. Для цього виконайте команду

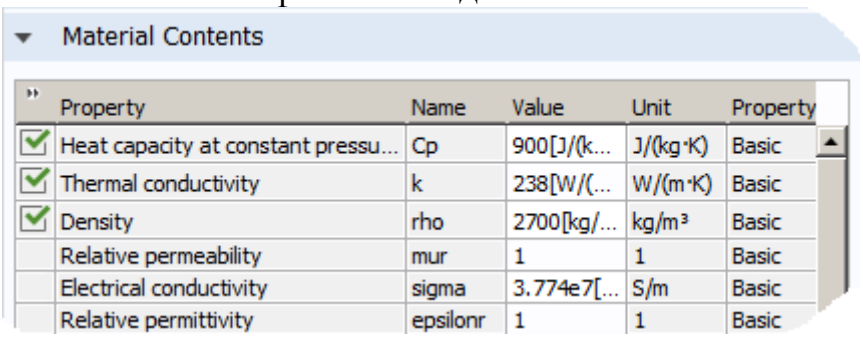
`Materials` -> Add Material

Праворуч від графічного вікна відкриється вікно `Add Material` (наступний рисунок праворуч). У вікні `Add Material` в дереві доступних матеріалів виберіть `Built-In – Aluminum`.

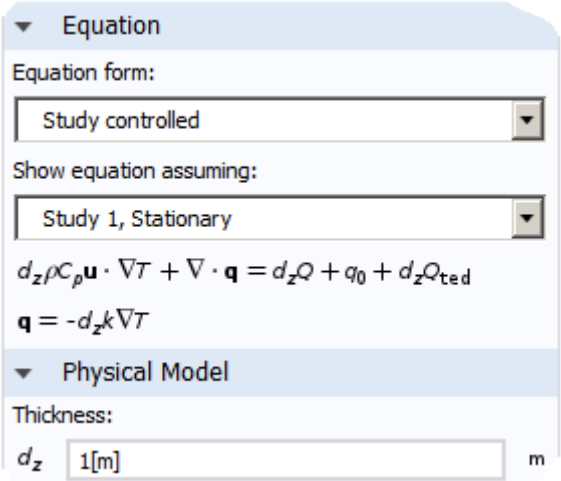




Клацніть по кнопці +Add to Component, яка розташована вгорі цього вікна, потім закрийте вікно «хрестиком» в правому верхньому кутку. В ДМ з'явиться вузол **Materials - Aluminum (mat1)** і його ПВ з таблицею фізичних параметрів. Галочки навпроти деяких властивостей вказують, що їх значення використовуються в поточній фізичної моделі.



Тепер клацніть по вузлу «фізики», в нашому випадку по вузлу **Heat Transfer in Solids (ht)**. В його ПВ ви побачите рівняння, які описують вибране фізичне явище.



Інтерфейс **Heat Transfer in Solids** використовується для розв'язання рівняння теплопровідності

$$d_z\rho C_{\text{P}}\frac{\partial T}{\partial t}+d_z\rho C_{\text{P}}\mathbf{u}\cdot\nabla T+\nabla\cdot\mathbf{q}=d_z\,Q,$$

де T - шукана функція температури, $\mathbf{q}=-d_zk\nabla T$ – тепловий потік, k - коефіцієнт теплопровідності, $\nabla=\left[\frac{\partial}{\partial x},\frac{\partial}{\partial y}\right]$ – оператор градієнта (в нашому випадку двовимірний). В рівняння входять матеріальні коефіцієнти: густина ρ, теплоємність C_{P}, і коефіцієнт теплопровідності k (скалярний або тензорний). В рівняння входять також вектор швидкості **u** речовини і джерела тепла Q різної природи. Для двовимірних задач використовується умовна товщина $d_z=1[m]$, яку не часто доводиться міняти.

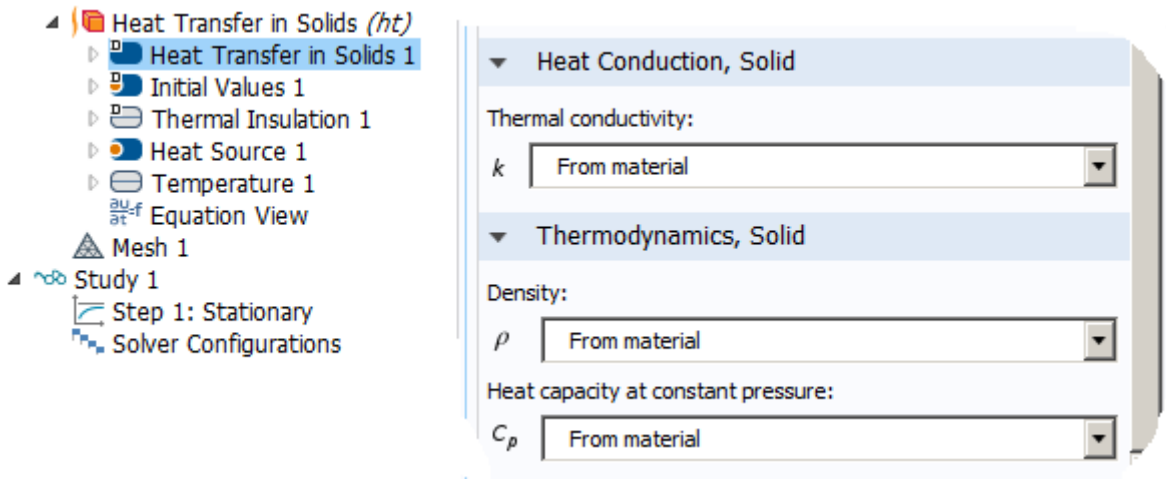
Перший член $d_z\rho C_\rho\frac{\partial T}{\partial t}$ в статичних задачах зникає. Для поля швидкостей **u** і джерел тепла слід створювати власні вузли ДМ. Якщо таких вузлів немає, то відповідні параметри вважаються рівними нулю. В нашому прикладі ми створимо одне джерело тепла потужністю Q_0, яке буде розташовано в області лівого кола. Тому рівняння насправді матиме вигляд $-d_z\,k\left(\nabla\cdot\nabla T\right)=d_zQ_0$, і параметр d_z не буде впливати на розв'язок.

Наведемо позначення, які використовуються під час запису рівнянь в вікнах «фізики».

$\nabla=\left[\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right]$	оператор градієнта
$\nabla u=grad\left(u\right)$	градієнт скалярної функції u
$\nabla\cdot\mathbf{u}=div\left(\mathbf{u}\right)$	дивергенція вектор-функції $\boldsymbol{u}$
$\nabla\times\mathbf{u}=rot\left(\mathbf{u}\right)$	ротор вектор-функції $\boldsymbol{u}$

Тут же в ПВ вузла **Heat Transfer in Solids (ht)** в розділі **`Domain Selection`** вказується індикатор (номер або номери) області, в якій буде розв'язуватися задача. Якщо геометрія складається з декількох підобластей, то можна вибрати всі або тільки частину з них. В ПВ вузла фізики внизу є розділ **`Discretization`**, в якому вибирається гладкість базисних функцій МСЕ, використовуваних при апроксимації розв'язку. Ви можете вибрати `linear` (лінійну), `quadratic` (квадратичну) або більш високу ступінь. Для більшості «фізик» за замовчуванням використовується квадратична апроксимація. У розділі **`Dependent Variables`** задаються імена змінних шуканих величин. Зазвичай для задач теплопровідності це `T`, а для математичних задач - `u`. Ви можете вказати інше ім'я.

Коефіцієнти рівняння задаються в ПВ вузла **Heat Transfer in Solids - Heat Transfer in Solids 1**. Якщо в цій панелі нічого не змінювати, то вони автоматично будуть взяті з параметрів матеріалу.



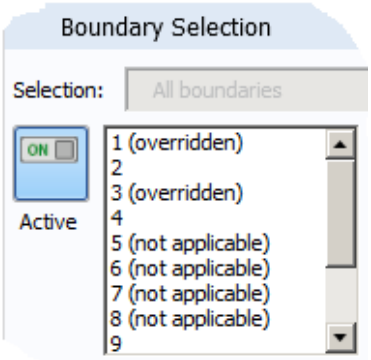
Граничні значення за замовчуванням задаються у вузлі **Zero Flux** і, як правило, зображають нульовий потік (тепла, речовини і т.д.) через границю. В фізичних інтерфейсах цьому вузлу зазвичай дається інша назва. В нашому прикладі автоматично створений вузол **Thermal Insulation 1** (теплоізоляція) задає нульовий тепловий потік на всіх ділянках границі. В більшості випадків вам буде потрібно це змінити. Для цього необхідно додати нові вузли, що представляють інші типи граничних умов, вказати в їх секціях вибору індикатори ділянок границі, на яких вони діють, і задати відповідні значення.

Задамо на лівій і верхній стороні прямокутника сталу температуру 293.15 [K]. Для цього виконайте команду

`Physics->Boundaries->`Temperature

Активуйте створений вузол **Temperature 1**, натисніть кнопку Active в розділі **`Boundary Selection`** (якщо вона ще не активна) і в графічному вікні клацніть по лівій і верхній сторонам прямокутника. Їх індикатори (номери) 1 і 3 додадуться в секцію вибору **Selection**. В розділі **`Temperature`** в полі $\mathbf{T_0}$ залиште значення `293.15 [K]`.

Якщо зараз відкрити ПВ вузла **Thermal Insulation 1**, то ви побачите, що для ділянок границі з номерами 1 і 3 гранична умова «теплоізоляція» перевантажена (overridden).



Вузол **InitialValues** створюється автоматично і містить початкові (зазвичай нульові) наближення для нелінійного розв'зувача стаціонарних задач, або початкові умови для динамічних задач. В статичних задачах вузол слід використовувати, коли початкову умову не можна припускати нульовою, наприклад, коли температура з'являється в рівнянні у вигляді `1/T`. В такому випадку для початку ітераційного процесу слід використати інше початкове значення. В нашому прикладі за замовчуванням початкове значення задано температурою `293.15 [K]`. Залиште його без змін.

Рівняння теплопровідності може мати члени, які моделюють внутрішні джерела тепла. У нас в правій частині рівняння стоїть член $d_z \cdot Q$, де Q являє собою об'ємну густину внутрішніх джерел тепла. Щоб їх врахувати, потрібно створити відповідний вузол ДМ і задати в його ПВ параметри джерела тепла. В нашому прикладі зоною, в якій буде діяти внутрішнє джерело тепла, є ліве коло (для цього воно і створювалося).

Побудуйте джерело.

`Physics` **`->Domains`** – Heat Source

В ПВ створеного вузла **Heat Transfer in Solids (ht) - Heat Source 1** в секцію вибору помістіть індикатор лівого кола. Для цього в графічному вікні підведіть покажчик миші до кола. Якщо воно не змінить колір, то крутніть колесо миші і, коли коло змінить колір, клацніть по ньому. Номер 2 (індикатор кола) з'явиться в секції вибору. Потім введіть інші параметри джерела тепла:

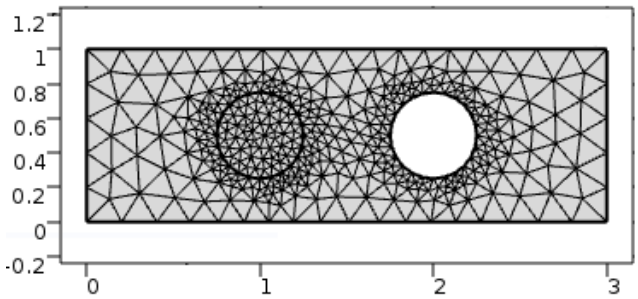
`Heat Source` – **General source** = `Включити` (*включити радіокнопку*)

`Heat Source` – **Q_o, < >** = `User defined, 100000`

Тут **Q_0** являє потужність джерела тепла (кількість енергії, що виділяється в одиниці об'єму за одиницю часу). Сумарна потужність джерела в нашому випадку дорівнює

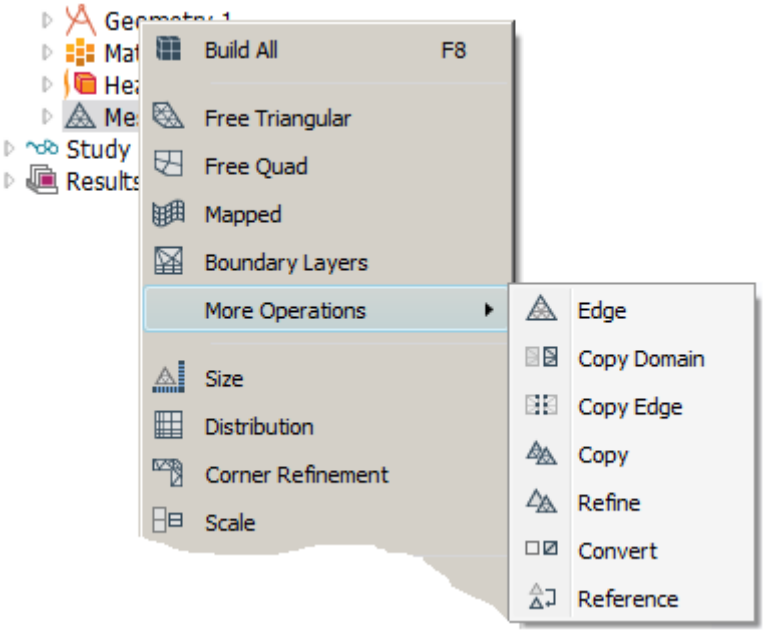
$$Q_\Sigma = Q_0 \cdot \pi r^2 = 100000[W/m^3] \cdot \pi \cdot (0.25[m])^2 = 19634.95[W/m].$$

Важливим елементом моделі, що істотно впливає на точність обчислень, є вузол **Mesh** (скінченно-елементна сітка). Є можливість вибирати обумовлену сітку або проектувати власну. За замовчуванням вибирається обумовлена (`Physics-controlled mesh`) сітка розміром `Normal`. Якщо активувати вузол **Mesh 1** і в його ПВ клікнути по кнопці Build All, яка розташована вгорі цієї панелі, то буде згенерована СкЕ сітка, яка показана на наступному рисунку.



В «серйозних» задачах починайте обчислення з сітки `Normal` і поступово робіть її більш дрібною до тих пір, поки розв'язок не перестане змінюватися.

Якщо натиснути правою кнопкою миші по вузлу **Mesh**, стануть доступними інструкції Free Triangular, Free Quad(rilateral), Mapped і Boundary Layers.

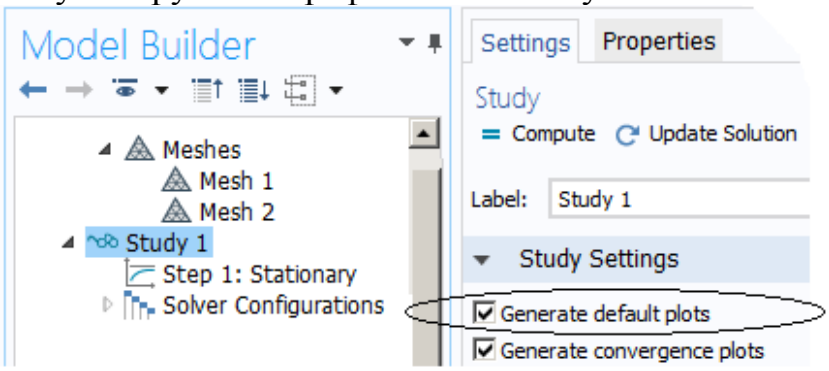


Перші дві покривають область (двовимірну) трикутною або чотирикутною сіткою. `Mapped` сітка покриває область чотирикутними елементами, які «дотримують» форму границі. Виберіть один з методів покриття, і виконайте команду Build All. Інструкція Refine дозволяє згустити існуючу сітку у всій розрахунковій зоні або тільки в деякій її підобласті. Для цього в ПВ вузла **Mesh** потрібно перейти в режим `User-controlled mesh`, а потім викликати команду Refine з контекстного меню або з закладки `Mesh` стрічки панелей. Залишимо в нашому прикладі «нормальну» сітку.

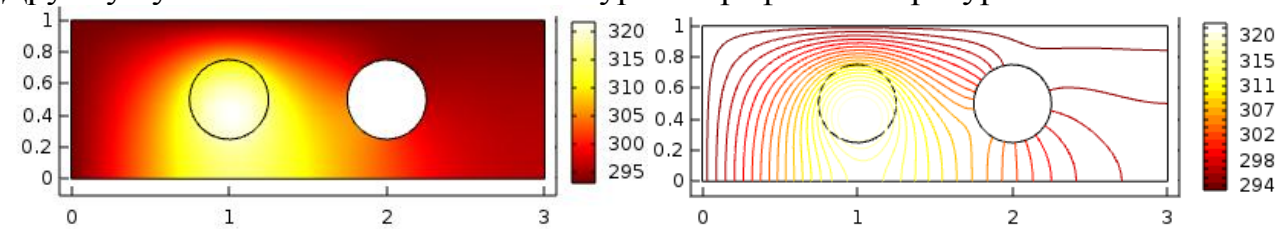
Обчислювання моделі виконується за командою Compute. Її можна викликати декількома способами: клацанням правою кнопкою миші по вузлу **Study 1** і вибором в контекстному меню пункту Compute, функціональною клавішею `F8`, або з панелі інструментів викликом інструкції `Home` – **`Study`** – Compute.

Здійсніть обчислення клавішею `F8`. У правому нижньому кутку головного вікна ви побачите індикатор виконання.

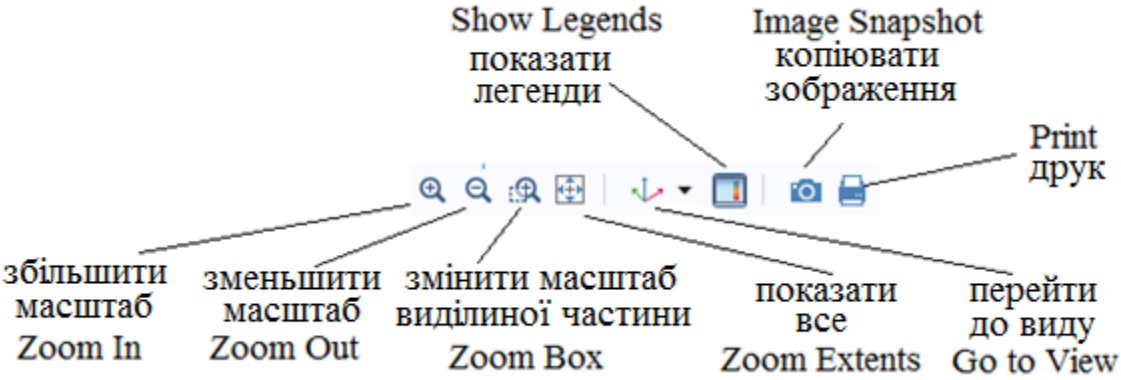
Якщо з якої-небудь причини виникне помилка, то з'являється вікно повідомлень. Крім цього, інформація про помилку записується в ПВ вузла **Study 1 - Solver Configurations - Solution 1 - Stationary Solver 1 - Information - Error 1**. Нагадаємо, що тут тире розділяє назви вузлів і підвузлів ДМ. Якщо помилок немає, то зазвичай автоматично створюються вузли з графіками розв'язків. В ПВ кожного вузла **Study** є прапорець **Generate default plots**, що встановлює ознаку генерування графіків за замовчуванням.



В нашому прикладі буде створено дві графічні групи (типу 2D Plot Group): **Temperature (ht)** і **Isothermal Contours (ht)**. Перша буде містити вузол **Surface 1**, що представляє графік температури з функціональним забарвленням. Друга у вузлі **Contour 1** містить контурний графік температури.



Над кожним графічним вікном розташована панель управління графіками. Її іконки дозволяють збільшити або зменшити зображення (Zoom In, Zoom Out, Zoom Box, Zoom Extents), зберегти його в файл чи буфер обміну (Image Snapshot), або надрукувати.

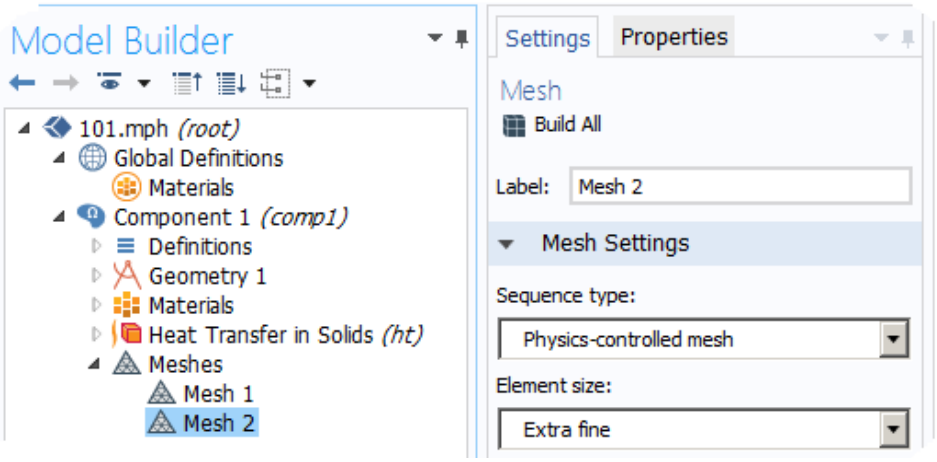


Двовимірні зображення можна змінювати за допомогою кнопок миші:

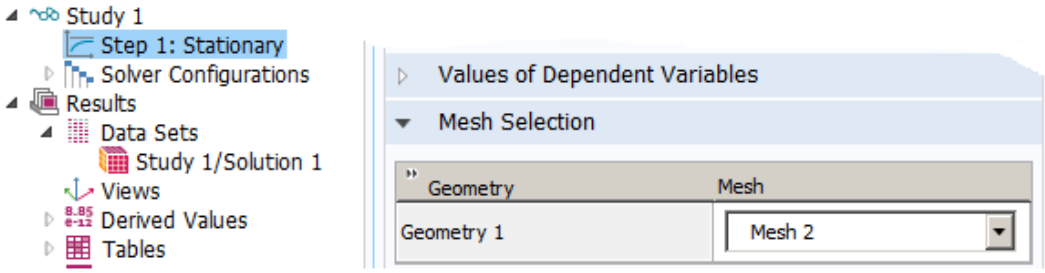
- натискання правої кнопки миші в графічній області і переміщення миші зсовує зображення;
- натискання середньої кнопки миші в графічній області і переміщення миші масштабує зображення;

Зображення можна повернути до стандартного виду, клацнувши по іконці Go to View, і вибравши в списку один із запропонованих «видів».

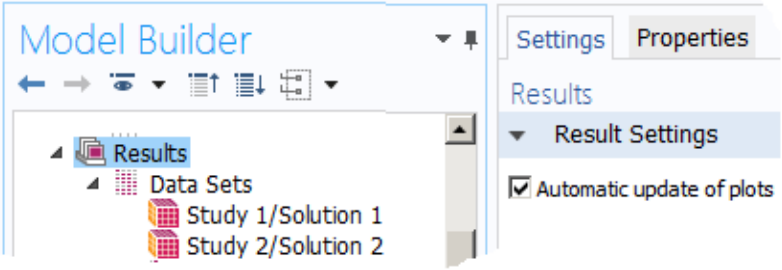
В компоненті можна мати кілька СкЕ сіток і виконувати обчислення для кожної з них окремо. Обидва попередніх графіка побудовані з використанням «нормальної» СкЕ сітки. Ви можете повернутися назад і створити другу більш дрібну сітку. Наприклад, клацніть правою кнопкою миші по вузлу **Component 1** і в контекстному меню виберіть команду Add Mesh. В гілці **Meshes** буде створений вузол **Mesh 2**, в ПВ якого ви можете задати Extra fine розмір нової сітки.



Після цього відкрийте ПВ вузла **Study 1 - Step 1: Stationary**, і в розділі **Mesh Selection** виберіть другу сітку Mesh 2. Повторіть обчислення моделі, натиснувши на клавішу F8.

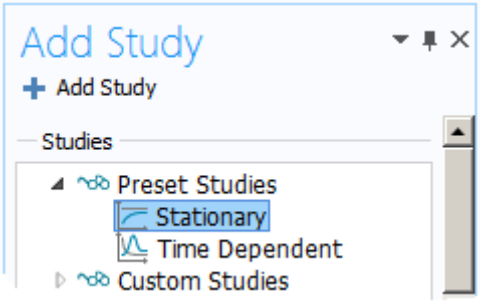


Результати розміщуються в наборі (джерелі) даних **Study 1/Solution 1**. Разом з цим будуть оновлені графіки **Temperature (ht)** і **Isothermal Contours (ht)**, які використовують ці дані. Автоматичне оновлення графіків задається установкою прапорця **Automatic update of plots** в ПВ вузла **Results**.



Вибирати різні сітки по черзі не завжди зручно. Якщо ви хочете, щоб обидва результати існували одночасно, то в моделі можна створити другий вузол **Study**. Для цього виділіть кореневий вузол моделі **root** (самий верхній вузол ДМ), клацніть по ньому правою кнопкою миші і в контекстному меню виберіть

команду Add Study. Система відкриє вікно вибору типу обчислення (Add Study). В нашому випадку це може бути стаціонарний або динамічний розрахунок.

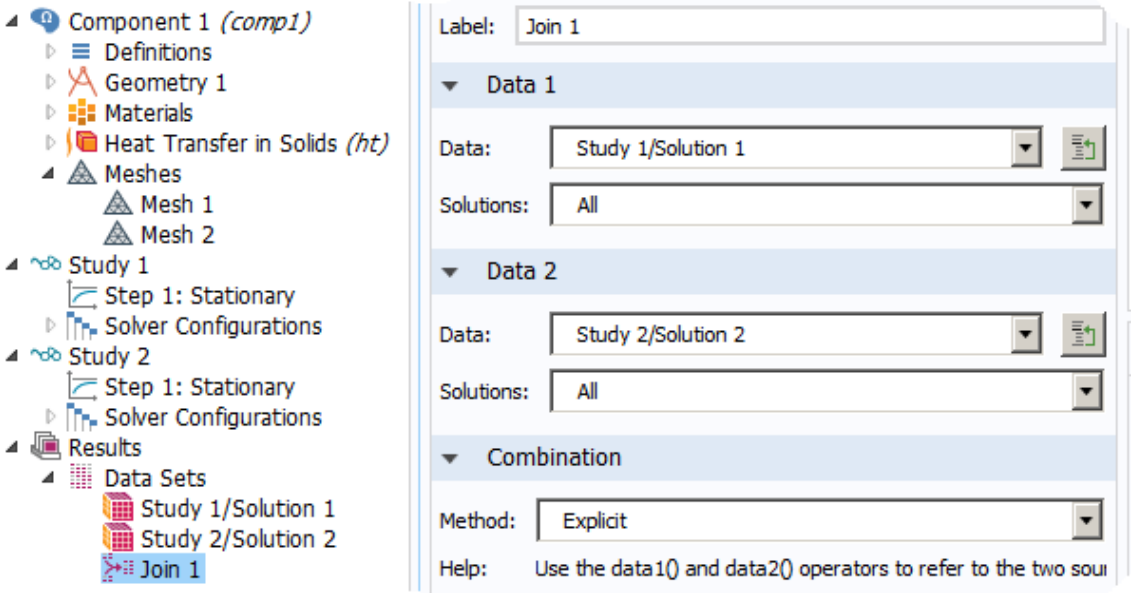


Виконайте подвійне клацання миші по рядку Stationary і закрийте вікно хрестиком в правому верхньому куті. В дереві моделі з'явиться вузол **Study 2**. Активуйте його підвузол **Step 1: Stationary**, і в ПВ в розділі **Mesh Selection** виберіть Mesh 2. Виконайте обчислення Study 2, наприклад, клавішею F8. Потім поверніться у вузлі **Study 1** до першої сітки Mesh 1 і перерахуйте цю версію задачі. Тепер в гілці даних **Results - Data Sets** будуть присутні два набори даних **Study 1/Solution 1** і **Study 2/Solution 2**, які представляють два розв'язки побудовані на різних сітках. Зверніть увагу, що кожен розрахунок **Study** будує по два графіка **Temperature (ht)** і **Isothermal Contours (ht)**. При цьому в назви графіків, побудованих після другого обчислення, будуть додані одиниці.

Нашою метою є побудова графіка різниці двох розв'язків. Для цього в гілці **Data Sets** створіть ще один набір даних, клацнувши правою кнопкою миші по вузлу **Data Sets**, і в контекстному меню вибравши команду Join. Тобто виконайте інструкцію

Data Sets: RMB > Join

Вузли типу Join дозволяють комбінувати розв'язки.



В ПВ створеного вузла **Join 1** в розділі **Data 1** виберіть перший розв'язок, а в розділі **Data 2** - другий (попередній рисунок праворуч). У розділі **Combination** в списку **Method** виберіть Explicit. Це означає, що в полях

виразів, в яких будуть використовуватися ці дані, до першого набору даних можна звертатися за допомогою оператора `data1(змінна)`, наприклад, `data1(T)`. Аналогічно, звернення до другого набору даних буде виконуватися з використанням оператора `data2(змінна)`. Таким чином, щоб створити графік різниці двох розв'язків потрібно побудувати графік виразу `data1(T) - data2(T)`. Виконаємо це за допомогою наступних інструкцій.

`Results` **-> Plot Group ->**2D Plot Group

Results - 2D Plot Group 5:

`2D Plot Group 5->` Surface

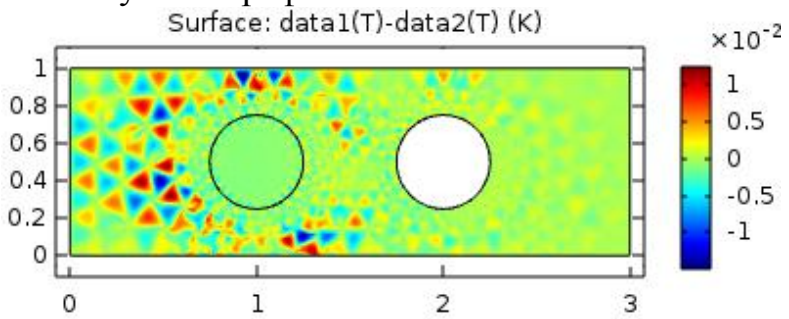
Results - 2D Plot Group 5 – Surface 1:

Data – **Data set** = `Join 1`

Expression – **Expression** = `data1(T)- data2(T)`

LMB > Plot

Буде побудований наступний графік.



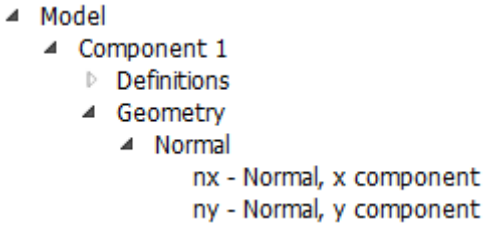
Зі шкали кольорів, розташованій праворуч, видно, що максимум різниці розв'язків приблизно дорівнює 0.015[K]. При цьому слід очікувати, що другий розв'язок точніший за перший, оскільки побудований на більш дрібній сітці.

Зауваження. Графік тієї ж різниці розв'язків можна було б побудувати дещо по-іншому. Для цього можна було б створити такий же вузол `Join`, але в списку **Combination** – **Method** можна вибрати значення `Difference` (Різниця). Використання виразу `T` в вузлі **Results - 2D Plot Group 5 - Surface 1**, який «спирається» на цей новий набір даних **Join 2**, означало б обчислення різниці температур першого і другого розв'язків.

При обчислюванні фізичних величин, таких як тепловий потік, потрібно записувати математичні вирази, в які входять імена різноманітних змінних. Деякі імена ми вже використовували. Наприклад в нашій задачі `T` являє температуру. Такі імена створюються разом з вибором прикладного режиму, в нашому випадку `Heat Transfer in Solids`. Імена `Tx` і `Ty` використовуються для позначення частинних похідних скалярного поля температури. Але в прикладних режимах створюється багато інших змінних, які можна використовувати в виразах.

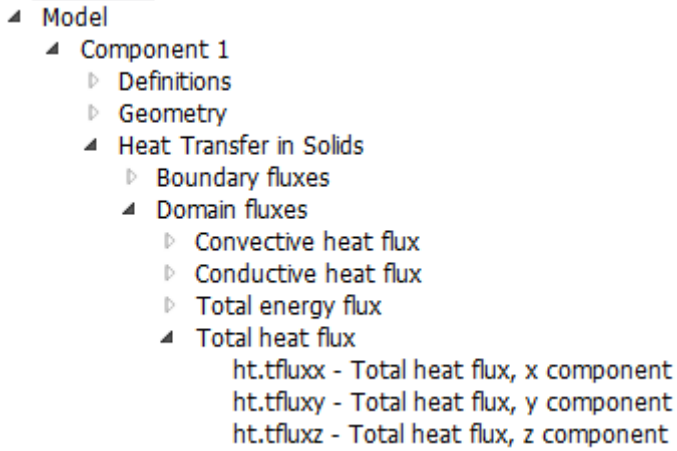
Деякі змінні COMSOL Multiphysics доступні для використання незалежно від діючого прикладного режиму. Серед них є змінні, які пов'язані з геометричними властивостями об'єктів моделі. Наприклад в двовимірних задачах ім'я `s` являє параметр вздовж кривої, а імена `tx,ty,nx,ny` представляють координати одиничних векторів дотичної і нормалі до кривої.

Коли задається вираз на кривій, наприклад гранична умава, то такі імена можна використовувати у відповідному полі **Expression**. Допустимі імена знаходяться в списку готових величин. Він відкривається кнопкою Replace Expression, яка розташована праворуч над кожним полем **Expression**. Наприклад, імена змінних, що представляють координати вектора нормалі кривої, знаходяться у списку/дереві готових величин в гілці `Model – Component 1 – Geometry –Normal`.



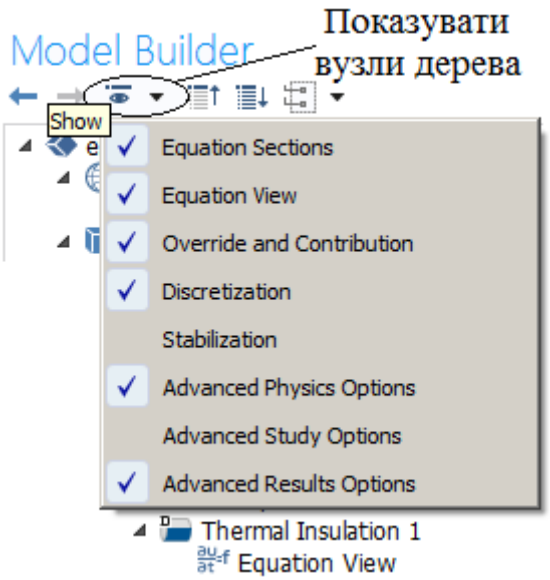
Припустимо, що ми бажаємо обчислити сумарний тепловий потік через межу джерела тепла (лівого кола). Для цього потрібно обчислити інтеграл $\oint_C \mathbf{q}\cdot\mathbf{n}\,ds$, де C – коло (крива інтегрування), ***q*** – вектор густини теплового потоку (тепловий потік віднесений до одиниці довжини), ***n*** – вектор нормалі до кривої.

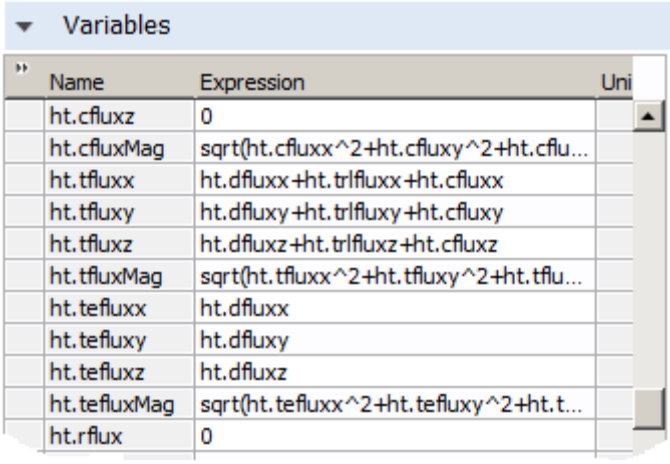
Позначення для координат вектора ***q*** можна знайти в тому ж списку/дереві вікна готових величин в гілці `Model – Component 1 – Heat Transfer in Solids – Domain fluxes – Total heat flux`.



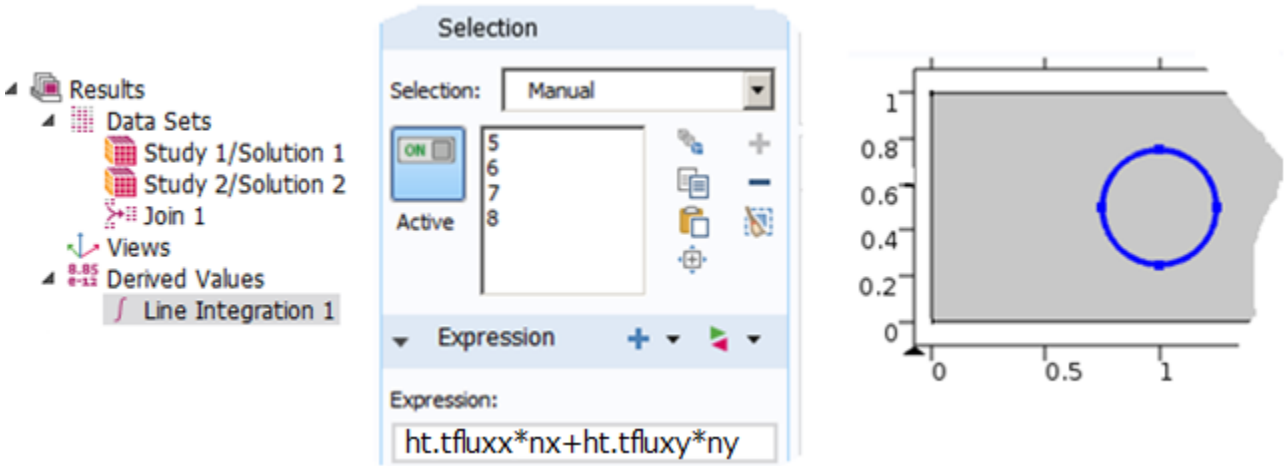
Отже імена `ht.tfluxx` і `ht.tfluxy` представляють координати вектора теплового потоку.

Інший спосіб знайти імена змінних, які визначені в обраному фізичному інтерфейсі, полягає в перегляді таблиці змінних в ПВ вузла **Heat Transfer in Solids 1 - Equation View**. У вузлах типу `Equation View` приводяться таблиці існуючих змінних моделі (наступний рисунок праворуч). Але такі вузли відображаються в ДМ лише тоді, коли в списку **Show**, який розташований вгорі панелі `Model Builder`, встановлено прапорець **Equation View** (наступний рисунок ліворуч).





Обчислимо сумарний тепловий потік через межу джерела тепла (ліве коло), тобто обчислимо інтеграл $\oint_C \mathbf{q}\cdot\mathbf{n}\,ds$. Для цього клацніть правою кнопкою миші по вузлу **Derived Values** і в контекстному меню виберіть команду Integration - Line Integration. В ПВ створеного вузла **Line Integration 1** в секцію вибору (клацаючи мишею в графічному вікні) помістіть індикатори дуг лівого кола 5, 6, 7, 8. Це означає, що інтегрування буде виконуватися по цим кривим (тобто по всій окружності). Щоб інтеграл обчислювався від величини $\mathbf{q}\cdot\mathbf{n}$, де $\mathbf{q}=(q_x,q_y)$ вектор теплового потоку, а $\mathbf{n}=(n_x,n_y)$ одиничний вектор нормалі до кривої, в поле **Expression** ПВ вузла **Line Integration 1** слід ввести вираз $q_x n_x + q_y n_y$. В позначеннях COMSOL Multiphysics він може буде записаний у вигляді `ht.tfluxx*nx + ht.tfluxy*ny`.



Клацніть вгорі панелі **Line Integration 1** по кнопці Evaluate, і в інформаційній зоні вікна програми з'явиться таблиця зі значенням `19583`. Тепер ще раз обчисліть інтеграл, але по лівій і верхній сторонах квадрата. Для цього видаліть з секції вибору номери дуг кола (повторно в графічній області клацніть по ним), і додайте індикатори `1` і `3` відповідних сторін прямокутника. Розкрийте кнопку - список Evaluate і виберіть `Table 1`. В тій же таблиці з'явиться

значення інтеграла 19611. Тепер інтеграл від того ж самого виразу обчислювався уздовж лівої і верхньої сторін квадрата. Оскільки повинен дотримуватися тепловий баланс, а на інших ділянках границі заданий нульовий тепловий потік, то обидва інтеграли повинні були б збігтися. Однак в силу наближеності обчислень вони дещо відрізняються. Повторіть два останніх обчислення, але в якості джерела даних для Line Integration оберіть **Study 2/Solution 2**. Помістіть результати в ту ж саму таблицю.

5,6,7,8 Study 1 (W/m)	1,3 Study 1 (W/m)	5,6,7,8 Study 2 (W/m)	1,3 Study 2 (W/m)
19583	19611	19583	19629

До речі, ми знаємо точну відповідь, бо сумарний тепловий потік через межу джерела тепла повинен дорівнювати сумарній потужності джерела, тобто величині $Q_{\Sigma} = 19634.95[W/m]$, яку ми обчислили раніше.

Загальну потужності джерела тепла ми можемо також обчислити як подвійний інтеграл по області. Дійсно, потужність внутрішніх джерел тепла ми задали при введенні даних у ПВ вузла **Heat Source 1**. Тому інтегрування цієї величини по області лівого кола дасть потрібну сумарну потужність.

Results -> Derived Values:

RMB > Integration - Surface Integration

Results -> Derived Values – Surface Integration 2:

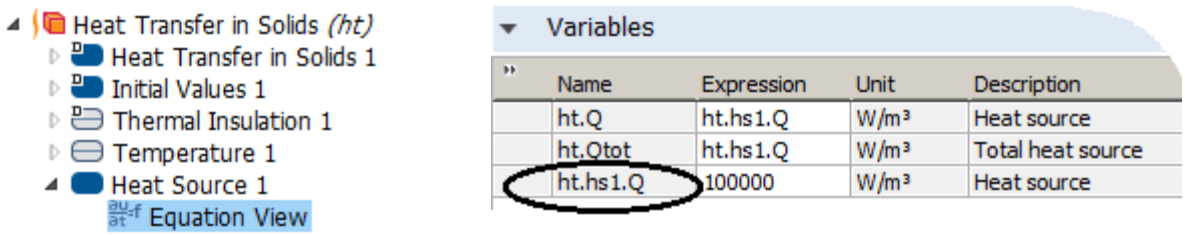
Data – **Data set** = Study 1/Solution 1

Selection – **Selection** = Manual

У розділі **Selection** натисніть кнопку Active (якщо вона ще не активна) і в графічній області клацніть по лівому колу, додавши індикатор 2 в секцію вибору. Це означає, що інтегрування буде виконуватися по цій області (тобто по всьому колу). Потім введіть:

Expression – **Expression** = ht.hs1.Q

Ім'я змінної ht.hs1.Q, яка представляє потужність джерела тепла, можна знайти у вікні **Equation View** вузла **Heat Source 1**.



Тепер надрукуємо значення інтеграла, виконавши команду

LMB > Evaluate –Table 1

В таблиці **Table 1** з'явиться значення 19635.

Взагалі, прикладні інтерфейси містять багато додаткових обчислюваних величин. Звісно, що загальна потужність джерел тепла теж обчислюється. Перевіримо це.

Для того, щоб друкувати значення, які не залежить від просторових координат, в гілці **Derived Values** створюються вузли типу `Global Evaluation`.

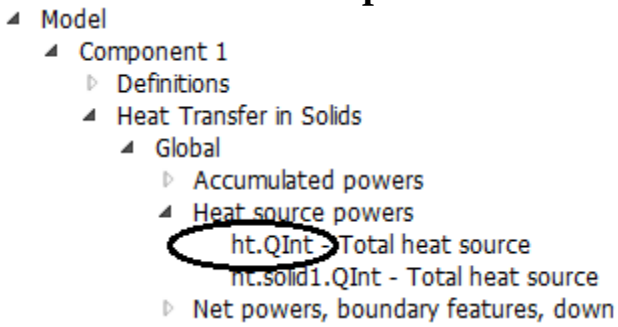
Results -> Derived Values:

RMB > Global Evaluation

Results -> Derived Values – Global Evaluation 1:

`Expression` – **Expression** = `ht.QInt`

Ім'я змінної `ht.QInt`, яка представляє загальну потужність джерела тепла, ми знаходимо у вікні готових величин поля **Expression**.



Надрукуємо це значення в першій таблиці.

LMB > Evaluate –Table 1

ht.tfluxx*nx +	(W/m) ht.tfluxx*nx +	ht.tfluxx*nx	..) ht.tfluxx*nx +	Heat source (W/m)	Total heat source (W)
19583	19611	19583	19629	19635	19635

Єдина відмінність від попередніх значень полягає в одиницях вимірювання, яка для змінної `ht.QInt` дорівнює `W` (вона вже помножена на 1 м умовної товщини конструкції).

Припустимо тепер, що ви бажаєте побудувати графік температури уздовж більшої осі прямокутника. Тоді в модель в гілку **Results** потрібно додати одновимірну графічну групу і лінійний графік. Але спершу потрібно створити джерело даних, яке зберігає інформацію про розв'язок уздовж бажаної лінії. Отже виконайте інструкції

`Results` –> Data Set - Cut Line 2D

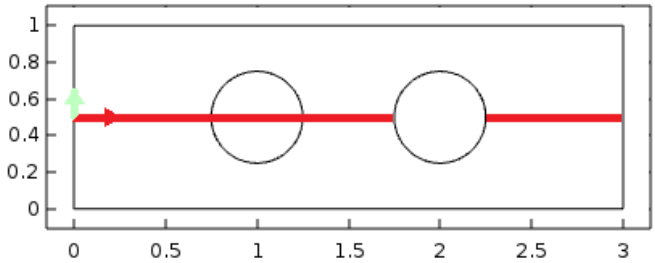
В ПВ створеного вузла **Cut Line 2D 1** введіть:

`Data` – **Data set** = `Study 1/Solution 1`

`Line Data` – **`Point 1`** – **x, y** = `0,0.5` (*два поля*)

`Line Data` – **`Point 2`** – **x, y** = `3,0.5` (*два поля*)

Побудуйте лінію командою Plot.



Тепер побудуйте на цій лінії графік. Створіть групу одновимірних графіків командою

`Results` –> 1D Plot Group

В ПВ створеного вузла **1D Plot Group 6** введіть:

Data – **Data set** = Cut Line 2D 1

Зверніть увагу, що на стрічці панелей з'явилася закладка 1D Plot Group 6. З неї виберіть команду

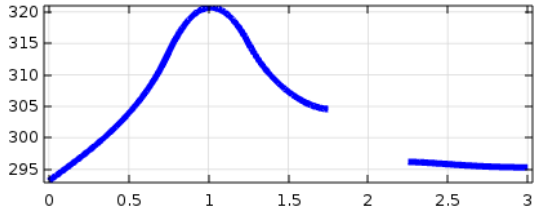
1D Plot Group 6-> Line Graph

В ПВ створеного вузла **Line Graph 1** введіть:

y-Axis Data – **Expression** = T

x-Axis Data – **Parameter, Expression** = Expression,x (*два поля*)

Побудуйте графік температури командою Plot.



Додайте сюди графік температури вздовж нижньої сторони прямокутника. Для цього в поточній графічній групі створіть ще один лінійний графік.

1D Plot Group 6-> Line Graph

В ПВ створеного вузла **1D Plot Group 6 - Line Graph 2** введіть:

Data – **Data set** = Study 1/Solution 1

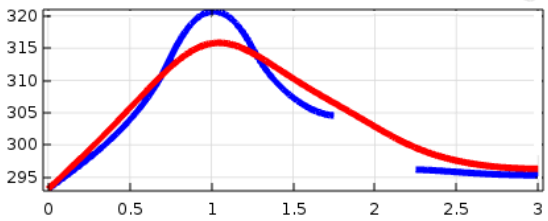
Selection – **Selection** = Manual

У розділі **Selection** натисніть кнопку Active і в графічній області клацніть по нижній стороні прямокутника, додавши індикатор 2 в секцію вибору. Потім виконайте інструкції:

y-Axis Data – **Expression** = T

x-Axis Data – **Parameter, Expression** = Expression,x (*два поля*)

LMB > Plot



Перед побудовою графіка вздовж лінії, якої немає в початковій геометрії, ми будували окреме джерело даних (Cut Line 2D 1). Для цього ми задали його положення в геометрії компонента. Але крім цього таке джерело містить всі дані з розв'язку Study 1/Solution 1, які відносяться до його геометрії. Після створення джерело даних можна використовувати для побудови графіків. Для цього в графічних вузлах в полях **Data** - **Data set** вказується ім'я цього джерела/набору даних. Для ліній, які вже присутні в геометрії компонента, створювати нове джерело даних немає потреби. Їх індикатори можна відразу розміщати (клацаючи по ним мишею) в секції вибору ПВ графічного вузла.

Якщо вам потрібно виконати інтегрування по лінії, якої немає в геометрії, то таким же чином ви повинні створити набір даних типу Cut Line 2D, а потім використати його в якості джерела даних для інтегрування. Якщо крива інтегрування присутня у вхідній геометрії, то створювати джерело даних не

потрібно. Таку криву можна відразу вибирати в якості джерела даних в ПВ вузла інтегрування.

Схожий підхід використовується в 3D випадку. Якщо розв'язок визначений в тривимірній області і треба проінтегрувати його по поверхні, якої немає в області, то слід створити джерело даних типу Cut Plane 3D і використати його для обчислення поверхневого інтеграла. Те саме джерело можна використовувати для побудови графіка функції на поверхні.

Аналогічно використовуються «точкові» набори даних Cut Point, які крім свого місця розташування містять інформацію про розв'язок задачі в точці (або точках). Значення розрахункових величин в точках, які вже є в геометрії, можна обчислювати відразу, вказавши точку в секції вибору вузла **Point Evaluation**. Якщо потрібних точок в геометрії немає, то слід побудувати джерело даних **Cut Point**, задавши в його ПВ положення точок і вказати розв'язок (якщо в моделі їх декілька) з якого будуть вибиратися дані.

Створіть точковий набір даних, який показаний на наступному рисунку. Для цього виконайте інструкції:

`Results` -> Data Set - Cut Point 2D

В ПВ створеного вузла **Cut Point 2D 1** введіть:

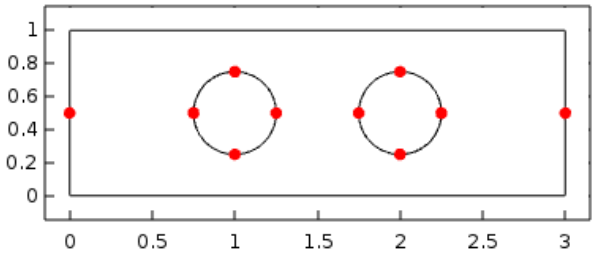
`Data` – **Data set** = `Study 1/Solution 1`

`Point Data` –**Entry method** = `Coordanates`

`Point Data` – **x:** = `0,0.75,1,1,1.25,1.75,2,2,2.25,3`

`Point Data` – **y:** =`0.5,0.5,0.75,0.25,0.5,0.5,0.75,0.25,0.5,0.5`

Тут в полях **Point Data-x** і **Point Data-y** вводяться абсциси і ординати точок набору даних **Cut Point 2D 1**. Побудуйте зображення точок командою Plot.



Тепер в точках, представлених в наборі даних `Cut Point`, можна обчислювати всі величини, які є в розв'язку **Study 1/Solution 1**.

У вузлі **Derived Values** створіть вузол типу `Point Evaluation`, який призначений для обчислення значень в точках. Для цього виконайте команду

Derived Values: RMB > Point Evaluation

В ПВ створеного вузла **Point Evaluation 1** введіть:

`Data` – **Data set** = `Cut Point 2D 1`

`Expression` – **Expression** = `T`

LMB > Evaluate – New Table

В гілці **Tables** з'явиться вузол нової таблиці **Table 2**, а сама таблиця буде відкрита в інформаційній зоні під графічним вікном. Вона буде містити значення температури в обраних точках. Якщо відкрити ПВ таблиці, то в розділі **`Column Headers`** ви побачите таблицю заголовків, яку можна

редагувати. Поміняйте заголовки так, як показано на наступному рисунку, залишивши тільки координати точок.

Column Headers

Column	Header
1	(0, 0.5)
2	(0.75, 0.5)
3	(1, 0.75)
4	(1, 0.25)
5	(1.25, 0.5)
6	(1.75, 0.5)
7	(2, 0.75)
8	(2, 0.25)
9	(2.25, 0.5)
10	(3, 0.5)

В результаті таблиця набуде наступного вигляду.

Messages Progress Log Table 2

(0, 0.5)	(0.75, 0.5)	(1, 0.75)	(1, 0.25)	(1.25, 0.5)	(1.75, 0.5)	(2, 0.75)	(2, 0.25)	(2.25, 0.5)	(3, 0.5)
293.15	313.48	308.84	319.13	315.06	304.57	294.63	302.78	296.16	295.29

Вузли `Point Evaluation` можна використовувати не тільки для друку значень розв'язку, але і для визначення значень фізичних параметрів. Для цього в геометрії моделі можна вибрати точку і вивести в ній значення будь-якого параметра. В нашому прикладі ви можете виконати наступні інструкції

Results - Derived Values:

RMB > Point Evaluation

В ПВ створеного вузла **Point Evaluation 2** в розділі **`Selection`** виберіть точку області і клацанням миші по ній додайте її номер (наприклад, 3 - крайня ліва точка лівого кола) в секцію вибору. Оскільки точка вже існує в геометрії, то створювати додаткове джерело даних не потрібно. Справа над текстовим полем **Expressions** розкрийте кнопку Replace Expression. У відкрившомуся вікні, розкрийте дерево змінних готових до вставки, знайдіть і розкрийте гілку `Material properties`, і подвійним клацанням виберіть бажаний параметр матеріалу (наприклад, середнє значення коефіцієнта теплопровідності).

Expression

Replace Expression

Expression:

ht.kmean

Unit:

W/(m*K)

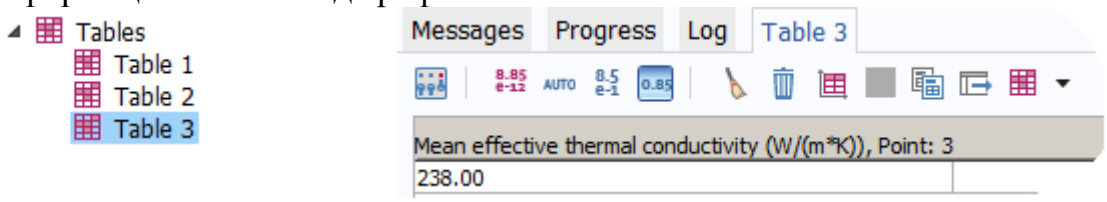
Type filter text

- Internal boundary fluxes, downside
- Energy
- Global
- Heat sources
- Material properties
  - Thermal diffusivity
  - Effective thermal conductivity
  - Thermal conductivity
  - ht.alphaTdMean - Mean thermal diffusivity
  - ht.C_eff - Effective volumetric heat capacity
  - ht.Cp - Heat capacity at constant pressure
  - ht.dz - Thickness
  - ht.gamma - Ratio of specific heats
  - ht.kmean - Mean effective thermal conductivity
  - ht.rho - Density

В полі **Expression** з'явиться вираз `ht.kmean`, значення якого можна відобразити в таблиці, якщо виконати команду

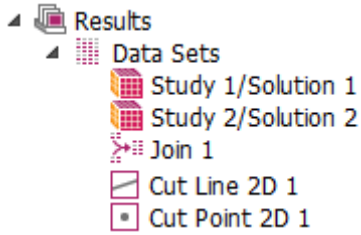
LMB > Evaluate – New Table

В гілці **Tables** з'явиться вузол нової таблиці `Table 3`, а сама вона буде відкрита в інформаційній зоні під графічним вікном.



Для ізотропного матеріалу коефіцієнт теплопровідності `k` зберігається у вигляді діагонального тензора з однаковими компонентами `kxx`, `kyy` і `kzz`, до яких у виразах можна звертатися за іменами `ht.kxx`, `ht.kyy` і т.д. Тут `ht` ім'я вузла **Heat Transfer in Solids (ht)**. У будь-якого вузла є назва (Label) і ім'я (Name). Тут `Heat Transfer in Solids` - назва (позначка) вузла, а `ht` - ім'я. Система автоматично надає імена деяким вузлам. В панелях властивостей таких вузлів в текстовому полі **Name** ці імена можна міняти. В інтерфейсі `Heat Transfer in Solids` скалярний коефіцієнт теплопровідності `k` обчислюється як середнє значення елементів `kxx`, `kyy` і `kzz`, і позначається через `ht.kmean`. Саме його значення відображено в таблиці на попередньому рисунку.

Як ви бачили, при обробці результатів обчислень важливу роль відіграють набори даних, що створюються в гілці **Data Sets**. Після обчислення моделі тут завжди присутній набір, який являє «повний» розв'язок (`Study 1/Solution 1`). Вузли типу `Study X/Solution X`, де `X` - цифра, як правило створюються солверами (розв'язувачами). Вони містять інформацію про компонент і його геометрію, про шукані величини, а також значення змінних, створених у вузлі **Variables** компонента.



Допоміжні набори/джерела даних створюються користувачем безпосередньо, або неявно через інші елементи ДМ. Наприклад, зонди (probes), якщо ви побудуєте їх в гілці **Definitions** компонента, самі створюють джерела даних.

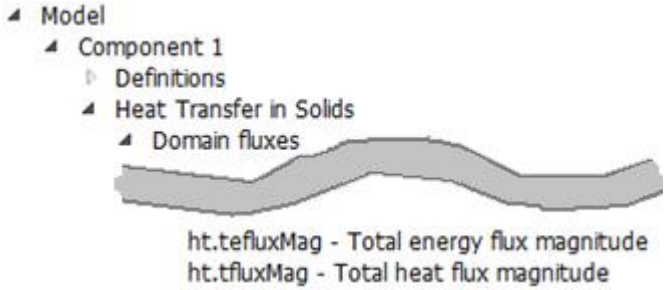
Крім гілки **Data Sets** з наборами даних, в гілці **Results** є вузол **Derived Values** (залежні величини). Він призначений для створення інструкцій обчислення залежних від розв'язку величин: мінімальних і максимальних значень, інтегралів, середніх значень, а також для обчислення окремих величин в точках.

Знайдемо максимальну величину теплового потоку. Спочатку виконайте команду

Results - Derived Values:

RMB > Maximum – Surface Maximum

В ПВ створеного вузла **Surface Maximum 1** в якості набору даних за замовчуванням використовується розв’язок **Study 1/Solution 1**. В секції вибору слід вказати індикатори 1 і 2 (вся геометрія), які визначають область де буде обчислюватися максимум. Вираз, максимум якого буде обчислюватися, вказується в полі **Expression**. Справа над текстовим полем **Expressions** розкрийте кнопку Replace Expression. У відкрившомуся вікні розкрийте дерево готових до вставки змінних, знайдіть і розкрийте гілку `Domain fluxes`, і подвійним клацанням виберіть бажану змінну `ht.tfluxMag`.



В полі **Expression** з'явиться вираз `ht.tfluxMag`, значення якого відобразиться в таблиці, якщо клікнути по кнопці Evaluate, розташованій вгорі ПВ вузла **Surface Maximum 1**.

Total heat flux magnitude (W/m^2)
18218

Тепер в ПВ вузла **Surface Maximum 1** поміняйте джерело даних, вибравши другий розв’язок **Study 2/Solution 2**. Знову знайдіть максимум. Для цього виконайте інструкції

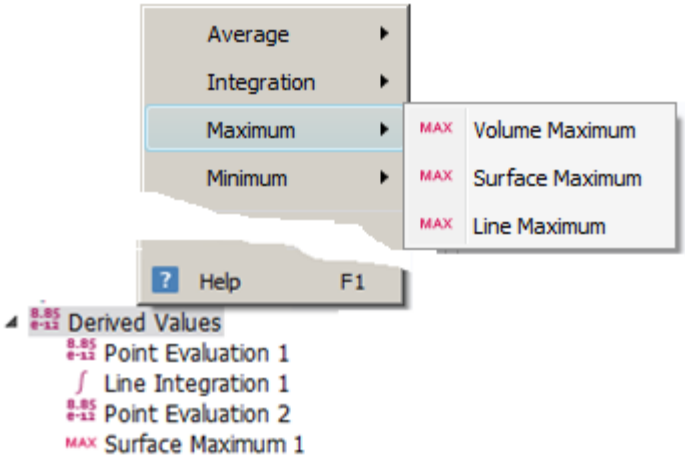
`Data` – **Data set** = `Study 2/Solution 2`

LMB > Evaluate – Table 4

У попередню таблицю додасться друге значення максимуму теплового потоку, але обчислене за розв’язком, побудованим на дрібнішій сітці.

Total heat flux magnitude (W/m^2)	Total heat flux magnitude (W/m^2)
18218	18056

Максимум і мінімум можна обчислювати на лініях, поверхнях і за об'ємом. Аналогічно обчислюються середні значення і інтеграли (по лініях, поверхнях і об'ємах). Це видно з контекстного меню, якщо натиснути правою кнопкою миші по вузлу **Derived Values** і розкрити підменю.



В гілці **Derived Values** знаходяться також вузли **Point Evaluation**, які використовуються для відображення значень змінних або виразів в конкретній точці (на відміну від максимуму, який обчислювався по області). Можна виводити значення в декількох точках одночасно і поміщати результати в одну таблицю. Вузли типу `Point Evaluation` ми створювали раніше.

Збережіть файл моделі, давши йому якесь осмислене ім'я, наприклад, Heat2D_InHoledRect.mph.

■

Файли моделей зберігаються у `*.mph` файлах командами File – Save або File – Save As.

Під час сеансу система зберігає історію роботи з моделлю і веде повний облік змін, які в ній чиняться. Це збільшує розмір файла. Щоб його стиснути існує команда File - Compact History (ущільнення історії), яка видаляє всі перевизначені зміни і скорочує розмір моделі. Перед остаточним збереженням моделі в файл рекомендується виконати цю команду.

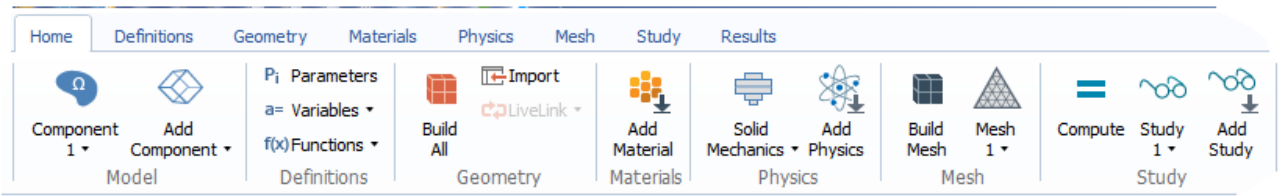
Підіб'ємо невеликий підсумок. Основним інструментом розв’язання задач в COMSOL Multiphysics є конструктор моделей (`Model Wizard`). Базовими об'єктами, які використовуються в конструкторі моделей, виступають вузли:

- **Global Definition** – призначений для визначення глобальних параметрів, змінних і функцій, які можна використовувати в будь-якому місці моделі;
- **Component** – призначений для створення основної моделі, включаючи геометрію конструкції, завдання матеріалів і розбиття на скінченні елементи; в гілці **Geometry** задається геометрична форма об'єкта, яка являє розрахункову область при чисельному моделюванні;
- **Study** – призначений для обчислювання задачі, вибору і налаштуванню параметрів розв’язувача (програми, яка реалізує чисельний алгоритм);
- **Results** – призначений для зображення та аналізу результатів, зокрема для побудови графіків розв’язків.

Фактично, вузли є інструкціями створення моделі. Послідовність вузлів утворює дерево моделі, а управління моделлю трансформується в роботу з деревом моделі. В наступному параграфі ми докладніше розглянемо деякі інструменти керування вузлами цього дерева.

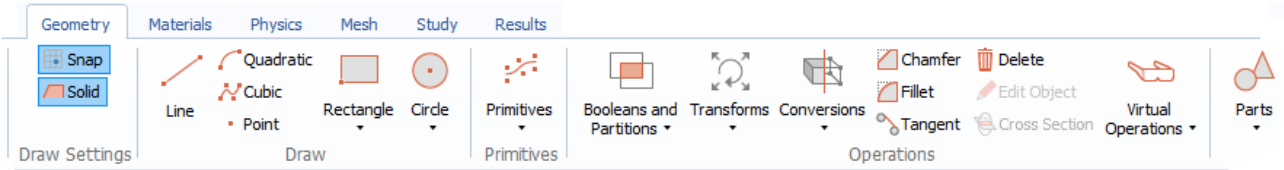
1.5.2. Керування деревом моделі

Головним інструментом управління деревом моделі COMSOL Multiphysics, починаючи з п'ятої версії, є стрічковий інтерфейс. Вважається, що стрічки дають більш наочний і простий спосіб управління програмами в порівнянні з вкладеними випадними меню. Філософія стрічки полягає в групуванні команд і відображенні тоді, коли їх можна використовувати. Команди групуються по вкладках, які в COMSOL Multiphysics впорядковані зліва направо відповідно до головної послідовності інструкцій створення моделей.



Інструментальна панель Home містить кнопки для загальних команд, таких як внесення змін до дерева моделі або виконання обчислення. Наприклад, якщо ви захочете змінити параметр, який управляє геометричним розміром, додати нову фізику, побудувати СкЕ сітку, виконати обчислення, додати новий графік - ви робите це з вкладки Home. Решта панелей інструментальної стрічки впорядковані зліва направо згідно потоку операцій, який заданий послідовністю розташування гілок в ДМ: Визначення, Геометрія, Матеріали, Фізика, Сітка, Дослідження та Результати.

Стрічка панелей реалізує три типи групування команд. Перший – стандартні закладки, які завжди відображаються в верхній частині програми. Вони відносяться до всіх основних етапів процесу моделювання і впорядковані відповідно до його основних етапів. На кожній такій закладці можна побачити всі команди, доступні для цього етапу. Наприклад, закладка Geometry містить значки команд, які можна використовувати при створенні геометрії моделі.

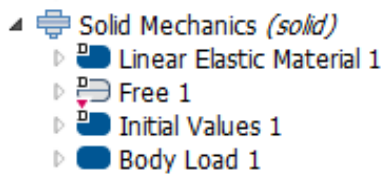


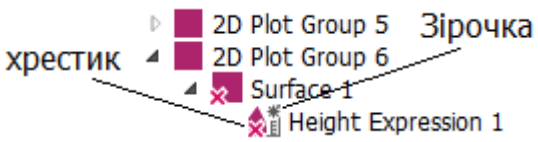
Другий тип - це контекстні закладки, які з'являються, коли ви працюєте зі специфічним об'єктом. Прикладом цього є закладки графічних груп, які з'являються на час роботи з графіками.

Третім типом логічного групування є модальні закладки, які з'являються, коли ви виконуєте дуже специфічні завдання, яких немає на стандартних закладках стрічки. Прикладом модальної закладки є Work Plane. Коли ви працюєте з «робочими перерізами» інші закладки недоступні, якщо їх команди не можуть керувати відповідними площинами. Наприклад, ви не зможете створити СкЕ сітку на робочій площині і тому закладка Mesh недосяжна.

Відмітною рисою кожного кроку роботи з моделлю є кольорова схема. Наприклад, всі значки панелі Geometry є червоно-помаранчевими, всі піктограми панелі Materials – жовто-оранжеві, всі значки панелі Physics – сині, а піктограми панелі Results – червоні. До того ж закладка Results має різні значки для різних розмірностей.

На піктограмах вузлів ДМ ви іноді будете бачити маленькі позначки. Маленький червоний трикутник на піктограмі вузла попереджає, що він (вузол) був переписаний; вузли іноді змінюють (переписують) вміст вікон властивостей інших вузлів, які в ДМ розташовані вище.

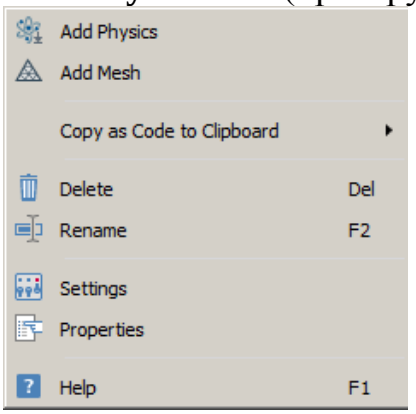


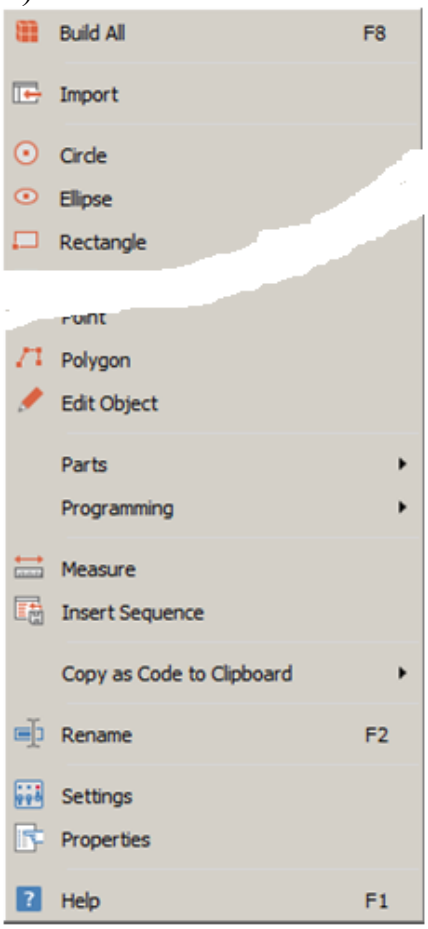


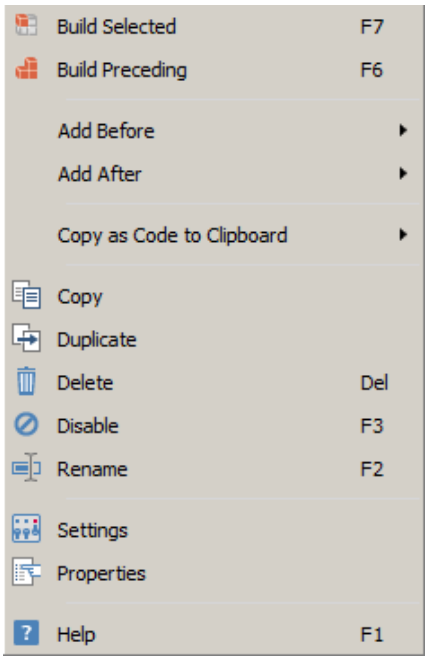
Буква D вгорі значка вузла означає, що вузол був створений системою автоматично з параметрами за замовчуванням. Зірочка в верхньому правому куті значка вказує, що вузол редагується. Червоний хрестик на піктограмі вузла означає, що є помилка в ПВ цього вузла.

При моделюванні відмінність між деревом моделі і стрічкою панелей полягає в тому, що ДМ показує нам те, що вже зроблено, а стрічка підказує, що може йти далі.

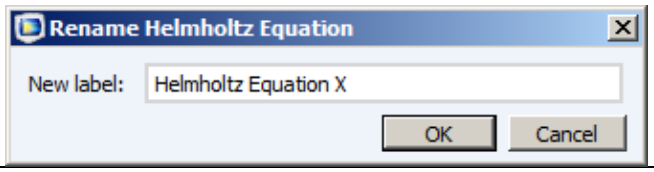
Майже кожну команду можна виконати двома способами (або навіть більшим): вибором іконки команди на стрічці або клацанням правої кнопки миші по вузлу ДМ і вибором команди з контекстного меню. Ці методи еквівалентні і доповнюють один одного. Розглянемо, як можна конструювати модель і управляти його деревом, використовуючи контекстне меню. Воно з'являється при натисканні правою кнопкою миші по вузлу дерева. Його зміст залежить від вузла, для якого воно викликано. На наступному малюнку показані контекстні меню гілки двовимірного компонента **Component** (ліворуч), його підгілки **Geometry** (в середині) і вузла графічного примітиву **Geometry - Circle** (праворуч).



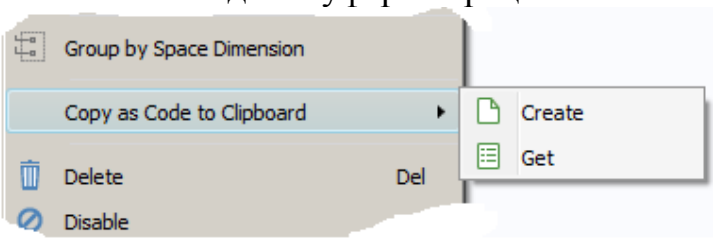




Ці меню сильно відрізняються і докладний опис команд, які можуть бути викликані з різних меню, фактично становить зміст нашого посібника. Однак є пункти, які зустрічаються майже всюди. Про них ми поговоримо в цьому параграфі.

Команда	**Опис**
Help	Відкриває контекстну довідку по поточному вузлу.
Properties	Відкриває службову закладку `Properties` вікна властивостей активного вузла.
Settings	Відкриває вікно властивостей вибраного вузла, тобто ту ПВ, в якій ми зазвичай міняємо налаштування і вводимо дані.
Rename	Відкриває вікно перейменування поточного вузла дерева моделі. 
Copy	Копіює вузол в буфер обміну. Після цього командою `Past назва_узла` можна вставити копію вузла в дерево моделі.
Duplicate (дублювання)	Вставляє в дерево моделі копію активного вузла. Дублікат вузла створюється нижче поточного в тій же гілці. Дублювання допустимо для вузлів багатьох типів, але не для всіх. Воно є зручним інструментом, оскільки виконує дії копіювання і вставки за один крок.
Delete	Видаляє поточний вузол з дерева моделі.
Disable	Деактивує поточний вузол, виключаючи його з послідовності інструкцій роботи з моделлю. Піктограма неактивного вузла в ДМ стає тьмяною.
Enable	Включає поточний вузол в послідовність інструкцій, якщо до цього він був деактивований. Тьмяна піктограма неактивного вузла повертається в початковий стан яскравості.
Move Up Move Down	Переміщують поточний вузол в ДМ на одну позицію вгору або вниз.

Команда Copy as code to clipboard використовується при програмуванні додатків. COMSOL Multiphysics дозволяє створювати застосування, що використовують його об'єктну модель. Команди підменю Copy as code to clipboard записують текст команди в буфер операційної системи.



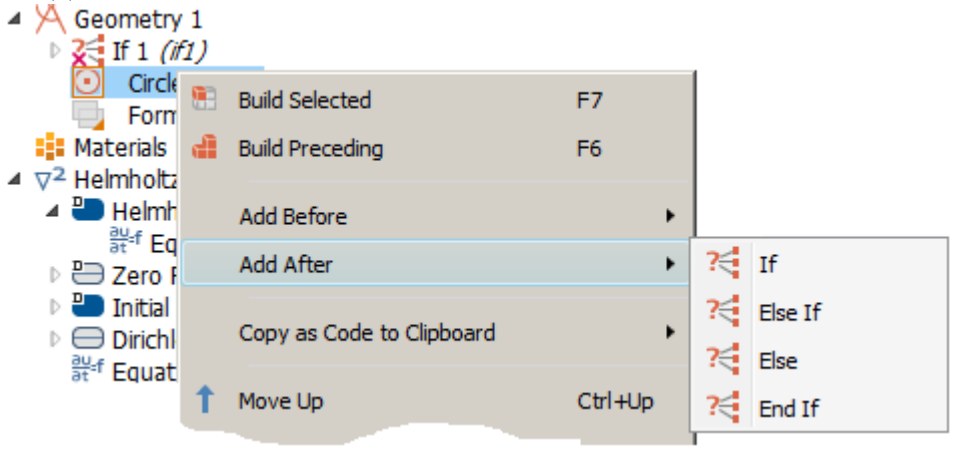
З буфера ви можете вставити текст коду в програму. Ось приклад коду, створеного командою Copy as code to clipboard - Create для вузла **Helmholtz Equation (hzeq)**, який вставлено з буфера в текстовий документ

```
model.physics().create("hzeq","HelmholtzEquation", "geom1");
```

Оскільки ми не маємо наміру описувати процес створення додатків, що використовують COMSOL API, то нам ця можливість не буде потрібна.

Команди Export та Import дозволяють експортувати/імпортувати дані в файли. Експортувати можна геометрію, анімацію, графіки, СкЕ сітку, таблиці, рівняння і деякі інші об'єкти. Імпортування виконується найчастіше з текстових файлів, які раніше створювалися для запам'ятовування даних. Все що експортувалося може бути імпортоване. Імпортувати можна також геометрію, створену в програмах сторонніх розробників, або створену в іншому файлі COMSOL Multiphysics.

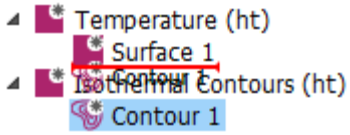
У вузлів, що представляють геометричні елементи, є підміню Add Before і Add After. Вони містять команди створення «умовних» послідовностей графічних вузлів/інструкцій. Це елементи, які включаються в послідовність геометричних вузлів, тільки при виконанні деяких умов, що накладаються на параметри моделі.



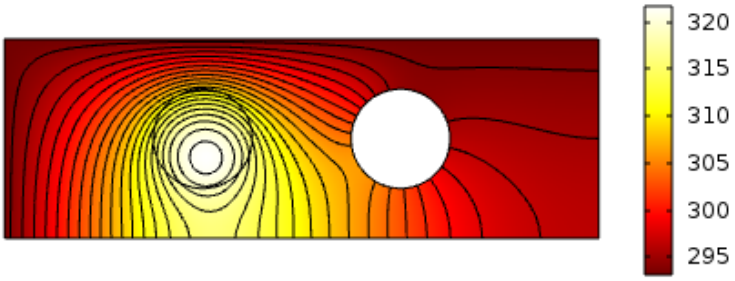
У розділі, що розповідає про створення геометрії, ми докладніше розглянемо приклади застосування таких умовних вузлів/інструкцій.

Всі операції в дереві моделі виконуються по черзі – зверху вниз. Фактично, вузли ДМ є інструкціями створення моделі, які повинні іти в певному логічному порядку. Таким чином, розташування вузлів в ДМ визначає порядок побудови і обчислення моделі. Але позиція вузлів в дереві моделі не фіксована. Багато з них можна переміщати. Однак таке пересування не повинно порушувати «розумну» послідовність. Наприклад, не можна поставити вузол граничних умов вище вузла «фізики», а вузол **Mesh** раніше вузла **Geometry**. Але місцезнаходження деяких вузлів не настільки важливо. Наприклад, не має значення порядок вузлів в гілці **Global Definitions**.

Графічні вузли можна переміщати в межах групи графіків або навіть переміщати в іншу графічну групу. Щоб, наприклад, перемістити графічний вузол, захопіть його лівою кнопкою миші (натисніть кнопку і не відпускайте) і перетягніть вниз або вгору в межах поточної графічної гілки або в іншу графічну групу того ж типу (наприклад, з **2D Plot Group** в таку ж саму). В процесі пересування позиція вузла в дереві позначається відрізком.



В прикладі попереднього параграфа (файл Heat2D_InHoledRect.mph) були створені вузли **Temperature (ht) - Surface 1** з функціональним забарвленням розрахункової області, і вузол **Isothermal Contours (ht) - Contour 1** з контурними лініями того ж температурного поля. Перенесемо вузол **Contour 1** в графічну групу **Temperature (ht)**. Для цього захопіть його лівою кнопкою миші і перетягніть вгору в графічну групу **Temperature (ht)** (обидві групи відносяться до типу 2D Plot Group). В переміщеному вузлі **Contour 1** внесіть зміни в колір контурних ліній (зробіть їх чорними) і відключіть виведення шкали кольорів (легенди). В результаті рисунок в графічній гілці **Temperature (ht)** набуде вигляду



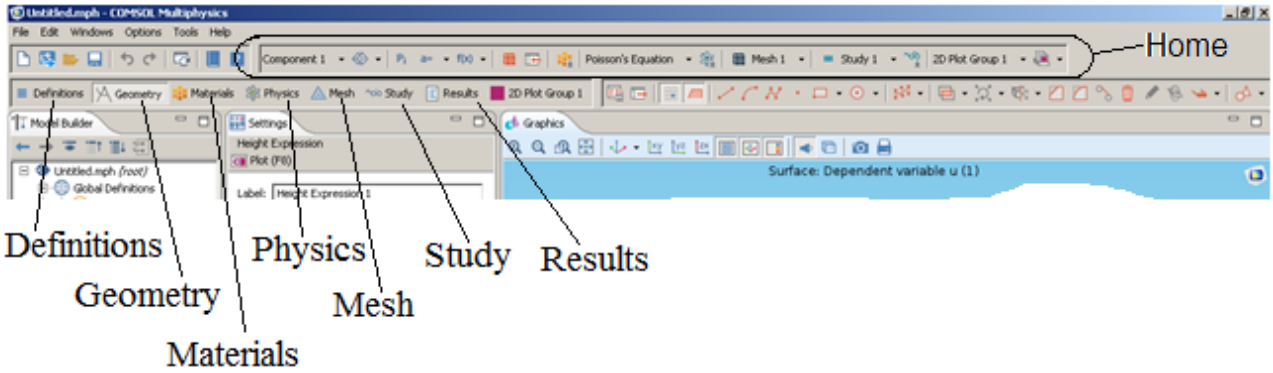
Порожню графічну гілку **Isothermal Contours (ht)** можна видалити. Для цього клацніть правою кнопкою миші по цьому вузлу і в випадному меню виберіть команду Delete.

Для керування деревом моделі і швидкого виконання деяких команд можна використовувати гарячі клавіші, а також поєднання клавіш клавіатури і миші. Наведемо деякі з них.

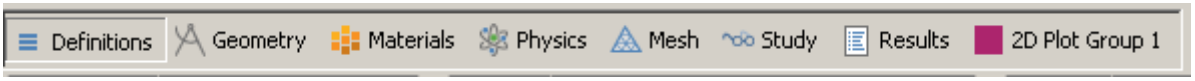
- `F1` – виклик контекстної довідки;
- `Ctrl-F1` – виклик довідкової документації;
- `F3` – відключає/деактивує виділений вузол дерева моделі;
- `F4` – включає вузол, який до цього був деактивований;
- `F7` – виконує завершальну команду поточного редагування вузла: один крок обчислювання (Compute Selected Step), будує елемент геометрії (Build Selected), будує ділянку СкЕ сітки (Build Selected);
- `F8` – повністю перебудовує геометрію (Build All), будує СкЕ сітку (Build All), виконує повне обчислення моделі (Compute), будує графік (Plot);
- `Shift+F10` – викликає контекстне меню;
- `Del` – видаляє поточний вузол ДМ;
- `CTRL – A` – виділяє всі об'єкти в графічному вікні;
- коли активна стрічка інструментальних панелей, колесо миші переміщує фокус між закладками стрічки.

Зауваження про панелі інструментів Windows XP. Якщо вашою операційною системою є Windows 7 або новіша, то подальший текст цього параграфа ви можете пропустити.

У Windows XP відсутня стрічка інструментальних панелей. Тому зовнішній вигляд головного вікна COMSOL Multiphysics відрізняється від стандартного, який ми спостерігаємо в Windows 7 або Windows 10. Замість головної інструментальної панелі з'являється рядок меню і панелі інструментів, звичайні для додатків Windows XP. Дерево моделі, вікна властивостей, вікно повідомлень і графічне вікно практично не відрізняються. На наступному рисунку показана верхня частина головного вікна COMSOL Multiphysics, запущеного під управлінням Windows XP.



В правій частині першого рядка панелі інструментів розташована панель `Home`. Другий рядок панелі інструментів містить закладки, ідентичні іншим основним закладкам головної інструментальної панелі. При виділенні ярлика закладки, в цьому ж рядку праворуч відкривається відповідна панель інструментів. На наступному малюнку показана ліва половина панелі інструментів з виділеної закладкою `Definitions`.

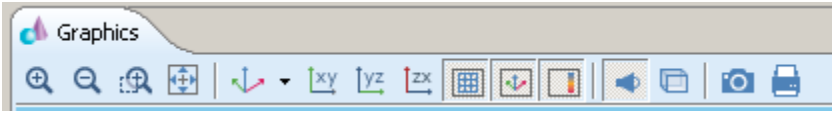


Права половина рядка містить кнопки команд, що відповідають цій закладці.

Закладки `Geometry`, `Materials`, `Physics`, `Mesh`, `Study` і `Results` також відображають панелі зі своїми кнопками команд.

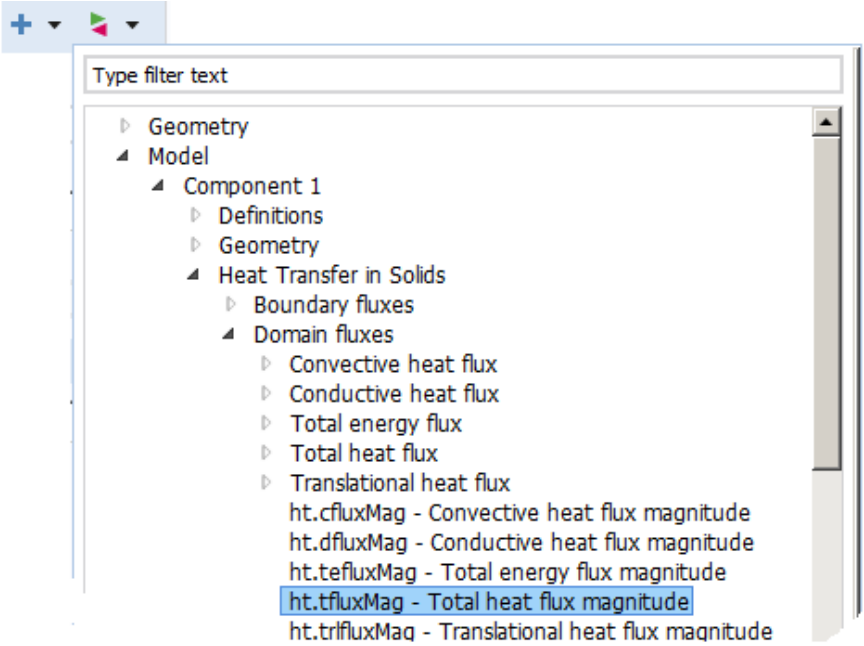
В Windows 7 (10) на стрічці можуть з'являтися контекстні закладки, що містять команди, залежні від виділеного в дереві моделі об'єкта. Аналогічно, в Windows XP можуть з'являтися контекстні панелі.

Панель команд, призначена для управління графічним вікном і яка розташована над ним, практично не відрізняється від аналогічної в Windows 7(10).

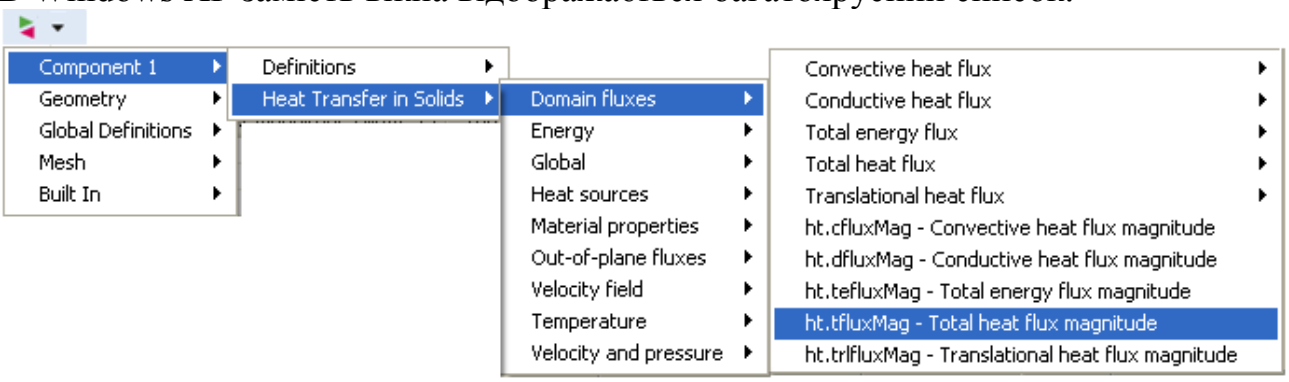


Невелика відмінність спостерігається при відображенні вікна готових величин для вставки в поля **Expression** (наступний рисунок).

Windows 7



В Windows XP замість вікна відображається багатоярусний список.



Зовнішній вигляд інших вікон мало відрізняється від стандартного.

Контекстні меню, які відкриваються клацанням правої кнопкою миші по вузлу дерева моделі, нічим не відрізняються від тих, які відкриваються в Windows 7.

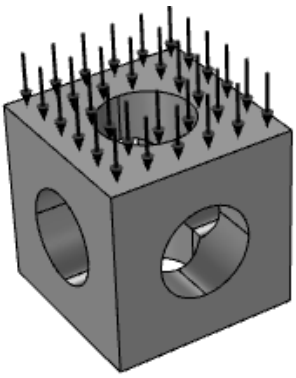
В нашому посібнику ми завжди будемо приводити зовнішній вигляд панелей і вікон системи COMSOL Multiphysics для Windows 7(10) . Якщо ви працюєте в Windows XP, то вікна будуть мати ту ж саму функціональність, але можуть мати трохи інший зовнішній вигляд.

1.5.3. Налаштування графіки.

Для керування зображеннями над кожним графічним вікном розташована панель управління. Її іконки дозволяють збільшувати або зменшувати зображення, зберігати його в файл чи буфер обміну (Image Snapshot), або друкувати.

Для більш ефективного управління графікою існують вузли, налаштування в яких впливають на те, що ми бачимо в графічних вікнах. Призначення таких вузлів та способи їх використання ми пояснимо, розглядаючи задачу деформування тривимірного тіла, яке показано на наступному рисунку. Воно становить бетонний куб $-1 \le x, y, z \le 1$ з циліндричними отворами радіусів

0.5 м, розташованими вздовж координатних осей. Основа куба закріплена. Зверху діє сила $F = 10000\, н/м^2$, спрямована вертикально вниз і рівномірно розподілена по поверхні грані.



Для розв'язання задач теорії пружності в COMSOL Multiphysics призначений прикладний інтерфейс Solid Mechanics. Використовуючи його, вам непотрібно записувати систему ДРЧП (програма зробить це сама). Треба лише побудувати геометрію тіла, вибрати матеріал, задати зовнішню навантагу та вказати обмеження/закріплення. В результаті система обчислює переміщення, напруження і деформації, а також багато інших параметрів.

Створіть модель послідовністю команд:

File->New->Model Wizard->3D->

Structural Mechanics - Solid Mechanics (solid)-> Stationary

Зверніть увагу, що шуканими є функції з іменами `u,v,w`, які були зазначені у вікні вибору фізичної моделі `Select Physics`.

В Comsol Multiphysics в виразах замість числових значень можна застосовувати параметри. Імена та значення параметрів задаються в таблиці параметрів ПВ вузла **Parameters**. В нашому прикладі ми створимо параметри, які представляють довжину сторони куба (a), радіус циліндричного отвору (r), величину зовнішнього тиску на верхню грань (F) і площу грані ($S = a^2 - \pi r^2$).

Задання параметрів моделі. Виконайте команду

Global Definitions : RMB > Parameters

В ПВ створеного вузла **Parameters 1** в колонці **Name** вводьте імена параметрів, а в колонці **Expression** – значення. Поля в стовпці **Value** будуть заповнюватися автоматично. В колонці **Description** вводяться коментарі. Заповніть таблицю параметрів так, як показано нижче.

Parameters

Name	Expression	Value	Description
a	2[m]	2 m	Сторона
r	0.5[m]	0.5 m	Радіус
F	10000[N/m^2]	10000 N/m²	Тиск
S	a^2-pi*r^2	3.2146 m²	Площа грані

Конструювання геометрії. При активному вузлі **Geometry 1** виконайте інструкції:

`Geometry`**->Pimitives->**Block

Geometry 1 – Block 1 (blk1):

`Size` – **Width** = `a`

– **Depth** = a
– **Height** = a
Position – **Base** = Center
LMB > Build Selected

Створіть три циліндри наступними інструкціями.

Geometry -> **Primitives** -> Cylinder

Geometry 1 – Cylinder 1(cyl1):

Size and Shape – **Radius** = r
– **Height** = a
Position – **x, y, z** = 0,0,-a/2 (*три поля*)
Axis – **Axis type** = z-axis

Geometry -> **Primitives** -> Cylinder

Geometry 1 – Cylinder 2(cyl2):

Size and Shape – **Radius** = r
– **Height** = a
Position – **x, y, z** = 0,-a/2,0 (*три поля*)
Axis – **Axis type** = Cartesian
Axis – **x, y, z** = 0,1,0 (*три поля*)

Geometry -> **Primitives** -> Cylinder

Geometry 1 – Cylinder 3(cyl3):

Size and Shape – **Radius** = r
– **Height** = a
Position – **x, y, z** = -a/2,0,0 (*три поля*)
Axis – **Axis type** = Cartesian
Axis – **x, y, z** = 1,0,0 (*три поля*)

LMB > Build All Objects

За допомогою операції логічного віднімання видалимо циліндри з області куба.

Geometry -> **Booleans and Partitions** -> Difference

Geometry 1 – Difference 1 (dif1):

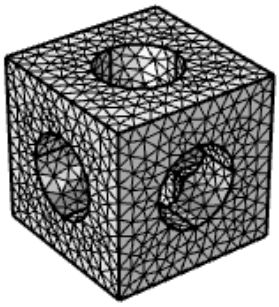
Difference – **Object to add** = blk1 (*куб*)
– **Object to subtract** = cyl1,cyl2,cyl3 (*циліндри*)

LMB > Build All Objects

В результаті буде побудований куб з циліндричними отворами, який показано на початку цього параграфа (без стрілок навантаги).

Побудуйте скінченно - елементну сітку з налаштуваннями за замовчуванням.

Component 1 – Meah 1: LMB > Build All

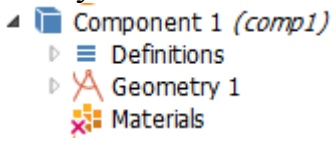


Вибір матеріалу. Виконайте команду

`Materials` **->** Add Material

Праворуч від графічного вікна відкриється вікно `Add Material`. У вікні `Add Material` в дереві доступних матеріалів виберіть `Built-In – Concrete` (бетон). Клацніть по кнопці +Add to Component, яка розташована вгорі цього вікна. Потім закрийте вікно «хрестиком» в правому верхньому куті. В ДМ з'явиться вузол **Materials - Concrete (mat1)** і його ПВ з таблицею фізичних параметрів. Галочки навпроти деяких властивостей вказують, що їх значення використовуються в поточній фізичній моделі.

Зауваження. Якщо в ПВ вузла «фізики» вказати вибір матеріальних параметрів `From material`, і спробувати обчислити модель без зазначення матеріалу, то в лівому нижньому куті піктограми вузла **Materials** з'явиться червоний хрестик (×), який позначає помилку.



Задайте навантаги і граничні умови, виконавши наступні інструкції.

`Physics` **->** **`Boundaries`** – Boundary Load

Boundary Load 1:

`Boundary Selection` – **Selection** = `4` (*верхня грань куба*)

`Force` – $\mathbf{F_A}$ = `0,0,-F` (*три поля*)

Оскільки зовнішній тиск `F` на верхню грань спрямований вниз (проти напряму вісі Z), то потрібно поставити знак мінус.

`Physics` **->** **`Boundaries`** – Fixed Constraint

Fixed Constraint 1:

`Boundary Selection` – **Selection** = `3` (*нижня грань куба закріплена*)

Крім граничних умов треба врахувати власну вагу тіла. Це об'ємна сила. Тому вона відноситься до групи навантаг, які задаються в випадному списку **`Domains`** панелі `Physics`.

`Physics` **->** **`Domains`** – Body Load

Body Load 1:

`Domain Selection` – **Selection** = `1` (*куб*)

`Force` – $\mathbf{F_V}$ = `0,0,-solid.rho*g_const` (*три поля*)

Тут $\mathbf{F_V}$ представляє вектор ваги одиниці об'єму. Оскільки цей вектор спрямований вниз, то ми поставили знак мінус. Зверніть увагу, як було використано ім'я густини бетону `solid.rho` – за допомогою точкової нотації. Стала `g_const` позначає прискорення вільного падіння.

Виконайте обчислення командою

`Home->`Compute

В гілці **Results** з'явиться гілка **Stress (solid)** тривимірної групи графіків з вузлом **Surface 1**, що містить поверхневий графіки напружень Мізеса.

Поверхневі графіки призначені для кольорового відображення значень величин на поверхнях тривимірних тіл.

Змінимо параметр, який відображається на поверхневому графіку.

Stress (solid) - Surface 1:

`Expression` – **Expression** = `solid.sz`

`Coloring and Style` – **Color table** = `RainbowLight`

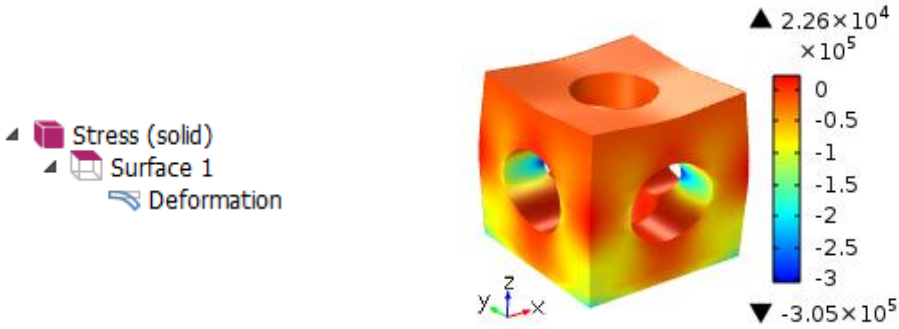
Тут `solid` являє ім'я «фізики» (ім'я, яке вказано в полі **Name** ПВ вузла **Solid Mechanics (solid)**), а змінна `sz` позначає σ_z напруження (внутрішню вертикальну силу, що припадає на одиницю площі перерізу нормального до вісі Z).

Для наочного зображення деформації система створила підвузол **Deformation**. У ньому в текстових полях розділу **`Expressin`** вказані імена координат вектора переміщення (`u,v,w`). Щоб краще побачити деформацію тіла ми лише змінимо масштабний коефіцієнт **Scale factor**. На зображенні це змінить зміщення кожної точки у зазначену кількість разів.

Stress (solid) - Surface 1 - Deformation:

`Scale` – **Scale factor** = `True`, `25000`

LMB > Plot



Справа від тіла зображено шкалу, яка представляє відповідність кольорів значенням напруження σ_z. Щоб на графіку не відображався контур тіла, а на шкалі кольорів відобразилося мінімальне і максимальне значення напруження σ_z, знімемо і встановимо відповідні прапорці:

Results - Stress (solid):

`Plot Settings` – **Plot data set edges** = `False`

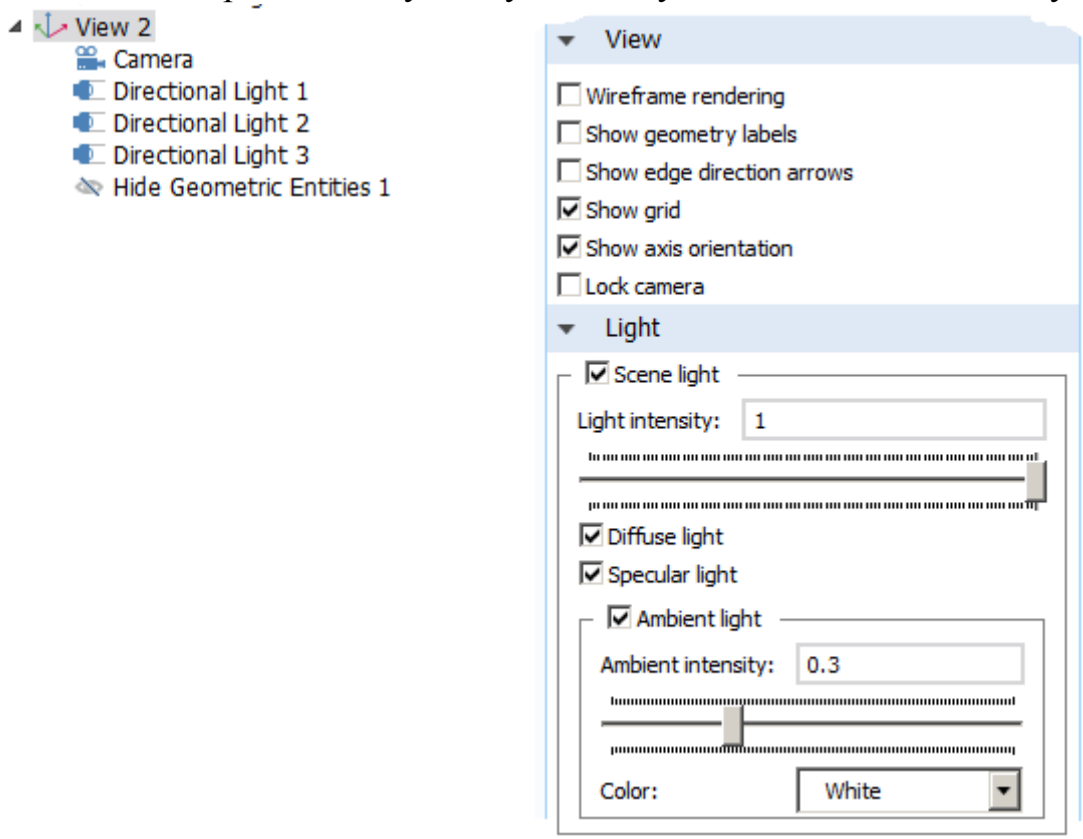
`Color Legend` – **Show mximum and minimum values** = `True`

LMB > Plot

Зверніть увагу на те, що хоча тіло стискається, максимальне (червоне) значення напруження σ_z додатне, тобто існують частини тіла, які розтягуються. Бетон добре витримує стискання, але погано працює на розтягнення. Отже в червоних частинах тіла можуть відбутися руйнування. Руйнування може відбутися і в стиснутій частині, якщо напруження стискання дуже велике. Щоб з'ясувати відбудеться руйнування чи ні, потрібно порівняти максимальні і мінімальні значення напруження з граничними значеннями для відповідної марки бетону.

Виникає питання, як роздивитися картину напружень σ_z на поверхні внутрішніх отворів. Для цього в системі існує інструмент, який дозволяє керувати відображенням частин геометрії і на графіку показувати не всі її елементи. Цим інструментом є вузли типу `View` (Вид). В них задаються

налаштування, які визначають спосіб відображення геометрії в графічних вікнах. У ПВ `View` вузлів встановлюються ознаки (прапорці) відображення службових елементів графіків: позначок геометричних елементів, координатної сітки і осей, спосіб освітлення і ступінь прозорості елементів геометрії. В тривимірних моделях у дочірніх підвузлах вузлів `View` визначаються джерела світла, налаштування камери спостереження і ознаки приховування деяких елементів геометрії. На наступному малюнку показана частина ПВ вузла `View`.



Зазвичай вузли `View` створюються в гілці **Definitions** компонента. Треба клацнути правою кнопкою миші по цьому вузлу і вибрати команду View з випадного меню. Один `View` вузол будується автоматично при створенні компонента, але таких вузлів може бути скільки завгодно.

У вузлі `View` автоматично створюються підвузли трьох спрямованих джерел світла. Всього в системі існує чотири типи джерел світла:

- спрямоване джерело (Directional Light) поводиться як світло, промені якого виходять з нескінченності;
- точкове джерело (Point Light) схоже на лампочку, яка світить на всі боки і ви можете задавати її положення;
- ліхтар (Spotlight); користувач може задавати положення джерела світла і його напрямок;
- фара (Headlight) прикріплена до «камери» і світить від неї (тобто завжди світить від вас).

Для додавання до виду/сцени джерела світла необхідно до вузла `View` додати підвузол відповідного типу.

У автоматично створюваному підвузлі `Camera` розташовані налаштування камери. Вони задають кут зору, місцезнаходження, масштаб і перспективу камери. Ви можете уявити собі справжню камеру, яка розташована перед графічним вікном, і ви дивитесь на об'єкт прямо скрізь видошукач камери. Модифікуючи параметри в ПВ вузла `Camera`, ви налаштовуєте камеру, змінюючи зображення, а не геометрію тіла.

Керувати камерою можна за допомогою кнопок миші:

- натискання правої кнопки миші в графічній області і переміщення миші зсовує зображення;
- натискання середньої кнопки миші (або колеса) в графічній області і переміщення миші масштабує зображення;
- утримування натиснутою клавішу `Ctrl` і однієї з кнопок миші, і переміщення миші змінює положення «точки спостереження/камери» і, як наслідок, змінює зображення (тільки для тривимірних зображень).
- утримування натиснутою клавішу `Alt` і однієї з кнопок миші, та переміщення миші обертає зображення навколо лінії зору.

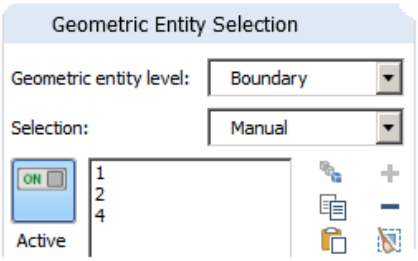
Рухаючись по графічному вікну ви зміщуєте камеру, а не саму геометрію.

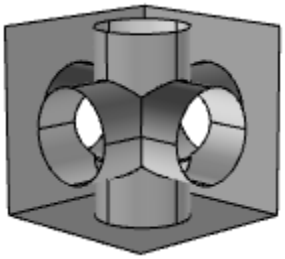
Для того, щоб сховати деякі елементи геометрії у вузлі **View** створюється дочірній вузол **Hide Geometric Entities**. Наприклад, спробуємо на попередньому зображенні сховати дві передні і верхню грань куба. Для цього потрібно створити «вид», в якому не будуть відображатися ці елементи геометрії. Виконайте команди

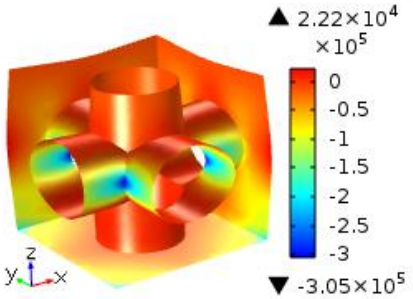
Component 1 – Definitions : RMB > View

Component 1 – Definitions – View 2 : RMB > Hide Geometric Entities

У ДМ створюється вузол **Hide Geometric Entities 1** (попередній рисунок ліворуч). В його ПВ в полі **Geometric entity level** виберіть `Boundary` рівень геометрії (щоб приховати елементи поверхні), і в секції вибору вкажіть індикатори `1`, `2`, `4` двох передніх і верхньої граней куба (наступний рисунок ліворуч). Для контролю можна клацнути кнопку Show Objects in Selection вгорі ПВ вузла **Hide Geometric Entities 1**. В графічній зоні буде побудовано зображення без зазначених поверхонь (наступний рисунок в центрі).







Перейдіть до вузла **Results - Stress (solid)** і в його ПВ введіть

`Plot Settings` – **View** = `View 2`

LMB > Plot

Поверхневий графік буде перебудований, і ви побачите зображення, яке показано на попередньому рисунку праворуч.

Отже налаштування в ПВ вузла **Hide Geometric Entities** дозволяють ховати деякі елементи геометрії (вони залишаються в геометрії, але не відображаються на графіках, в яких використовується батьківський «вид»). Зауважимо, що існує схожий вузол **Hide Geometric Objects** використання якого дозволяє ховати цілі об'єкти з геометрії компонента.

Щоб далі не було плутанини, поверніться у графічній групі **Stress (solid)** до попереднього виду.

Results - Stress (solid):

Plot Settings – **View** = View 1

LMB > Plot

Поверхневий графік відображає в кольорах величину на поверхні тіла. Якщо потрібно зобразити величину всередині тривимірного тіла, то для цього можна використати пошаровий (Slice) графік. Він показує розподіл величини всередині тіла, розфарбовуючи різними кольорами плоскі шари, які проходять скрізь тіло. Побудуємо один такий графік за допомогою наступних інструкцій.

Results->3D Plot Group

3D Plot Group 2:

Data– **Data set** = Study 1/Solution 1

Plot Settings – **View** = View 1

3D Plot Group 2 **-> Add Plot** – Slice

3D Plot Group 2 – Slice 1:

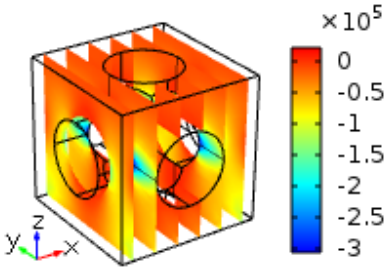
Data– **Data set** = From parent

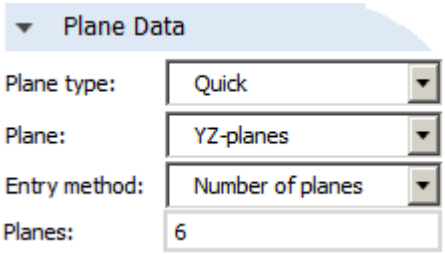
Expression – **Expression** = solid.sz

Plane Data – **Entry method, Planes** = Number of planes, 6

Coloring and Style – **Color table** = RainbowLight

LMB > Plot





Зверніть увагу, що в ПВ вузла **Slice 1** в розділі **Plane Data** зазначається кількість зрізів (поле **Planes**). Крім того, якщо встановлено прапорець **Interactive**, то за допомогою бігунка **Shift** можна переміщати всю «пачку» зрізів як тверде тіло.

Зняти з геометрії «верхній шар» і зазирнути всередину моделі можна також за допомогою вузлів типу Filter. Вони створюються як підвузли графічних вузлів і містять логічний вираз з критерієм, що визначає які частини графіка зображати.

Створимо ще одну тривимірну графічну групу з поверхневим графіком, на якому будемо відображати переміщення (модуль вектора переміщення), яке в інтерфейсі Solid Mechanics позначається змінною disp.

Results->3D Plot Group

3D Plot Group 3:

Data– **Data set** = Study 1/Solution 1

Plot Settings – **View** = View 1

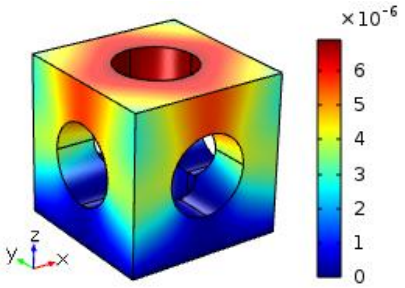
3D Plot Group 3 **-> Add Plot** – Surface

3D Plot Group 3 – Surface 1:

Data– **Data set** = From parent

Expression – **Expression** = solid.disp

LMB > Plot



Тепер додамо до графіка фільтр.

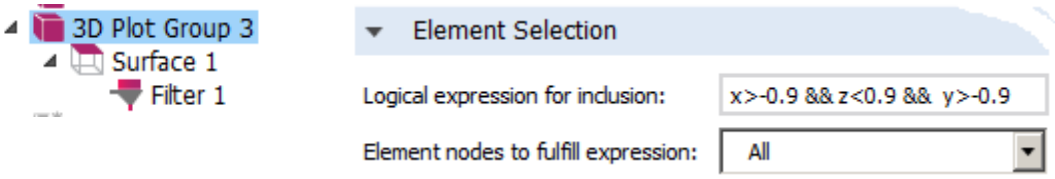
3D Plot Group 3 – Surface 1: RMB > Filter

3D Plot Group 3 – Surface 1 – Filter 1:

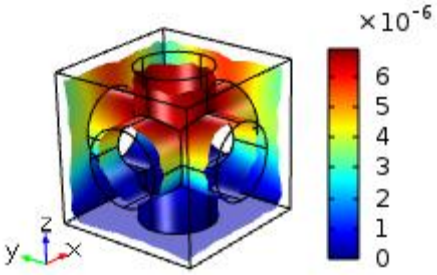
Element Selection – **Logical expression for inclusion** =

x>-0.9 && z<0.9 && y>-0.9

LMB > Plot



В результаті будується поверхневий графік величини переміщення, який показаний на наступному рисунку.



Тут логічний вираз x>-0.9 && z<0.9 && y>-0.9 «зняв» з геометрії «передній шар» і залишив на графіку всі елементи, координати точок яких задовольняють заданому критерію.

Можна запропонувати ще одне корисне застосування «фільтрів». Повернемося до поверхневого графіка **Results – Stress (solid)** напруження σ_z. Ми бачили, що в деяких точках тіла напруження додатне $\sigma_z > 0$, тобто бетон розтягується. Щоб він не руйнувався при розтягуванні, зазвичай розтягнуті ділянки бетону армують. Але з графіка **Stress (solid)** не досить зрозуміло, які частини тіла розтягуються, а які ні. Спробуємо на поверхні тіла виділити кольором частини тіла, які зазнають розтягування. Для цього накладемо на існуючий графік інший поверхневий графік, який за допомогою відповідного

фільтра буде відображати червоним (або іншим) кольором тільки ті частини поверхні, в яких напруження $\sigma_z > 0$ додатне. Для цього в нашу графічну групу додамо ще один `Surface` графік. Щоб він не відрізнявся по розташуванню від **Surface 1**, додамо для нього таку ж саму деформацію. Крім того, в якості виразу для забарвлення поверхні використаємо одиницю і для неї оберемо червоний колір. Якщо більше нічого не робити, то червоний графік закриє собою поверхневий графік напруження σ_z тому, що його вузол розташований в ДМ нижче. Але ми використаємо фільтр, який на новому графіку дозволить відобразити тільки частини поверхні з додатним напруженням $\sigma_z > 0$. В результаті закритою буде лише частина поверхні **Surface 1**.

Виконаємо інструкції, які реалізують описані побудови.

Results - Stress (solid): RMB > Surface

Results - Stress (solid) - Surface 2:

`Expression` – **Expression** = `1`

`Coloring and Style` – **Coloring, Color** = `Uniform, Red`

`Inherit Style` – **Plot** = `Surface 1`

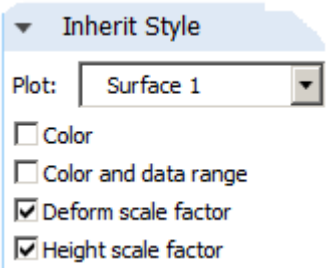
– **Color** = `False`

– **Color and data range** = `False`

– **Deform scale factor** = `True`

– **Height scale factor** = `True`

Пояснимо сенс полів розділу **`Inherit Style`** (успадкування стилю). Часто потрібно, щоб графіки, розташовані в одній групі, мали однакове кольорове оформлення і однакову деформацію. Ви можете в кожному з них задавати однакові параметри, але це клопітно. В ПВ багатьох графічних вузлів є розділ **`Inherit Style`** (успадкування стилю). Вибір в його списку **Plot** імені графічного вузла, означає використання параметрів оформлення цього вузла для поточного графіка. Графічний вузол може успадковувати стилі будь-якого з графіків його графічної групи, вузли яких розташовані вище в цій групі. Нижче списку **Plot** розташовані прапорці, які вказують, які елементи оформлення слід успадковувати.



Спадкування стилю також означає, що будь-які зміни оформлення в «батьківському» графіку автоматично призведуть до змін у всіх графіках, які успадковують його стиль.

В нашому прикладі ми відключили спадкування кольорової схеми і залишили спадкування деформацій. Вони успадковуються тільки тоді, коли для поточного вузла буде створено підвузол `Deformation`. Зробимо це.

Results - Stress (solid) - Surface 2:

RMB > Deformation

Залишилося додати фільтр. Для цього виконайте інструкції

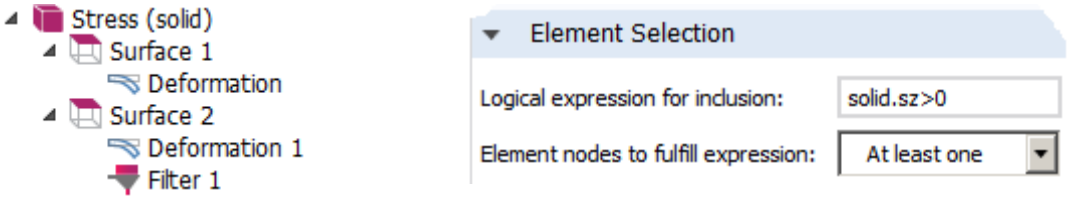
Stress (solid) - Surface 2:

RMB > Filter

Stress (solid) - Surface 2 – Filter 1:

`Element Selection` – **Logical expression for inclusion** = `solid.sz>0`

`Element Selection` – **Element nodes to fulfill expression** = `At least one`



Вибір в списку **Element nodes to fulfill expression** значення `At least one` додає в вибірку всі скінченні елементи, для яких хоча б в одному вузлі задовольняється критерій, заданий в логічному виразі. Це приводить до забарвлення скінченних елементів, які повністю або частково задовольняють критерію.

Завершіть побудову графіка клацанням по кнопці Plot (наступний рисунок ліворуч).

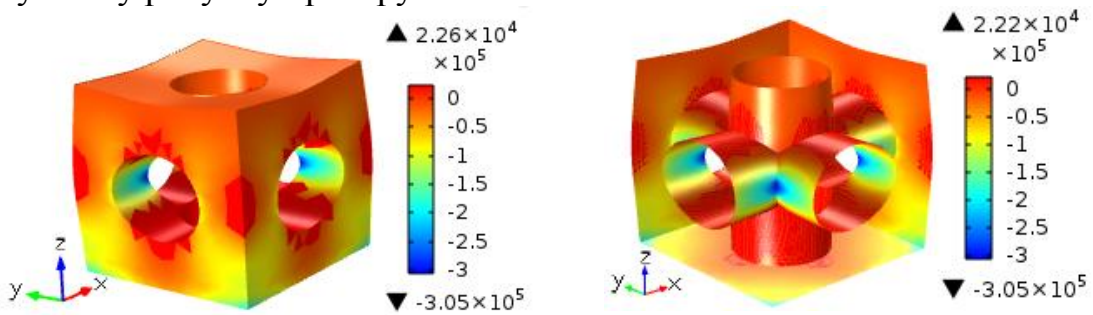
Тепер в ПВ вузла графічної групи **Stress (solid)** виберіть другий вид, в якому не відображаються три фронтальні грані.

Results - Stress (solid):

`Plot Settings` – **View** = `View 2`

LMB > Plot

Графік буде перебудований, і ви побачите зображення, яке показано на наступному рисунку праворуч.



Як бачите вид `View 2` було застосовано до обох графіків **Surface 1** і **Surface 2** графічної групи **Stress (solid)**.

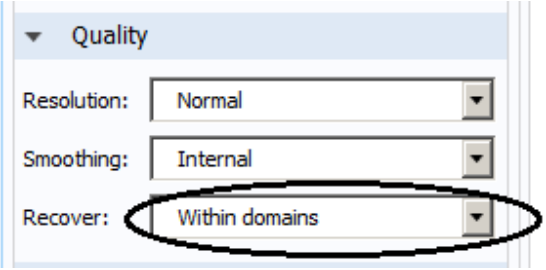
У вузлі **Stress (solid)** поверніться до попереднього виду:

Results - Stress (solid):

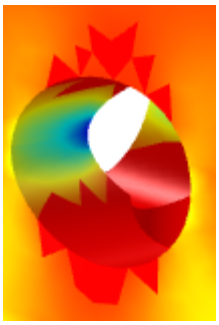
`Plot Settings` – **View** = `View 1`

LMB > Plot

В цьому місці доречно пояснити використання опції **Recover** (Відновлення) розділу **`Quality`** (Якість). У вузлі **Surface 2** розкрийте розділ **`Quality`** і в списку **Recover** виберіть `Within domains` (Всередині областей).



Виконайте команду Plot і графік трохи зміниться. Щоб побачити ці зміни на наступному рисунку зображено околицю лівого отвору попереднього графіка, розташованого ліворуч, з виключеною (`Off`) і включеною (`Within domains`) опцією **Recover**.



Під час розв'язання задачі методом скінченних елементів у вузлах сітки обчислюються значення якої-небудь однієї основної величини, наприклад, вектора переміщення (`u,v,w`). Інші величини, в нашому прикладі напруження, зазвичай обчислюються за допомогою диференціювання основної величини. При цьому в COMSOL Multiphysics використовується спеціальний поліноміальний алгоритм обчислення похідних високого порядку. Цей алгоритм відновлює похідні з використанням значень основної величини на деякій множині скінченних елементів. Однак при побудові графіків для скорочення часу обчислення похідні відбудовуються за значеннями розв'язку в одному скінченному елементі. Це призводить до того, що на графіках величини, обчислювані з використанням таких похідних, мають меншу точність, ніж розраховані в розв'язку. Наприклад, другі похідні дорівнюють нулю, коли в розрахунках використовуються лінійні елементи. Якщо використовувати поліноміальний алгоритм і значення розв'язку в вузлах декількох скінченних елементів, похідні відновлюються точніше (похідні відновлюються на основі множини елементів, а не в кожному елементі окремо). Оскільки ця техніка вимагає більше часу для обчислень, то за замовчуванням при побудові графіків вона відключена. В графічних вузлах вона включається при використанні опції **Recover** = `Within domains`. Це відновлює похідні в областях, не виконуючи згладжування розв'язку через границі.

Таким чином, в нашій моделі налаштування **Recover** = `Within domains` обчислює напруження на основі багатьох скінченних елементів, а не одного. Остання пара графіків показує, як може змінитися зображення при включенні опції відновлювання похідних.

Зауважимо, що в системі є оператор `ppr(...)`, який виконує ті ж самі дії, що і опція **Recover** = `Within domains`.

Відобразити зони додатного напруження можна простіше. Для цього достатньо побудувати графік виразу `solid.sz>0`. Це логічний вираз, який може набувати значення 1 (істина) або 0 (брехня). В областях, де його значення дорівнює 1, напруження розтягувальні.

Додамо нову 3D графічну групу, скопіюємо з групи **Stress (solid)** в буфер обміну вузол **Surface 1**, і вставимо його з буфера в створену графічну групу. В ній для вузла **Surface 1** відключимо деформування, і в полі **Expression** запишемо вираз `solid.sz>0`. Ці дії реалізуємо наступною послідовністю інструкцій:

`Results->`3D Plot Group

Results - Stress (solid) - Surface 1:

RMB > Copy

Results - 3D Plot Group 4:

RMB > Paste Surface

Results - 3D Plot Group 4 - Surface 1:

`Expression` – **Expression** = `solid.sz>0`

`Plot Settings` – **View** = `View 1`

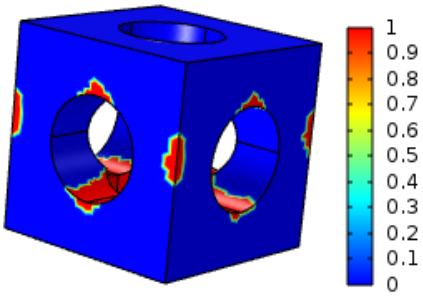
`Quality` – **Recover** = `Within domains`

Results - 3D Plot Group 4 - Surface 1 - Deformation:

RMB > Disable

Results - 3D Plot Group 4: LMB > Plot

В результаті буде побудовано наступний графік.



Тут ми ще включили опцію відновлювання похідних **Recover** = `Within domains`.

Розглянемо ще одне службове застосування вузлів `Deformation`. Припустимо, що ми бажаємо порівняти напруження σ_z в двох перерізах тіла $z=-0.95$ та $z=-0.5$. Це можна було б зробити, використовуючи джерела даних, які співпадають з цими перерізами.

Results – Data Sets: RMB > Cut Plane

Data Sets - Cut Plane 1:

`Data` – **Data set** = `Study 1/Solution 1`

`Plane Data` – **Plane** = `XY-planes`

`Plane Data` – **Z-coordinate** =`-0.95`

Results – Data Sets: RMB > Cut Plane

Data Sets - Cut Plane 2:

`Data` – **Data set** = `Study 1/Solution 1`

`Plane Data` – **Plane** = `XY-planes`

Plane Data – **Z-coordinate** =-0.5

Results->3D Plot Group

3D Plot Group 5:

Data– **Data set** = None

3D Plot Group 5 **-> Add Plot** – Surface

3D Plot Group 5 – Surface 1:

Data– **Data set** = Cut Plane 1

Expression – **Expression** = solid.sz

3D Plot Group 5 **-> Add Plot** – Surface

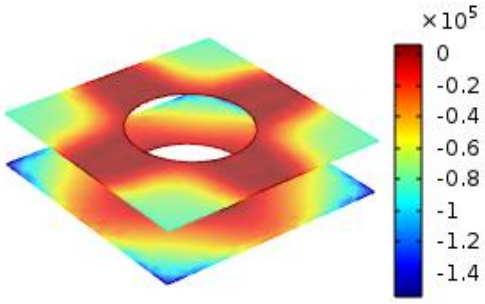
3D Plot Group 5 – Surface 2:

Data– **Data set** = Cut Plane 2

Expression – **Expression** = solid.sz

Inherit Style – **Plot** = Surface 1

LMB > Plot



Порівнювати зображення отриманих перерізів незручно. Це можна зробити краще, якщо розташувати перерізи поруч на одному графіку. Для цього треба побудувати їх зображення в спільній двовимірній графічній групі, і друге зображення зсунути відносно першого (без зсуву один переріз закриє інший). Зробимо це.

Results->2D Plot Group

2D Plot Group 6:

Data– **Data set** = None

2D Plot Group 6 **-> Add Plot** – Surface

2D Plot Group 6 – Surface 1:

Data– **Data set** = Cut Plane 1

Expression – **Expression** = solid.sz

2D Plot Group 6 **-> Add Plot** – Surface

2D Plot Group 6 – Surface 2:

Data– **Data set** = Cut Plane 2

Expression – **Expression** = solid.sz

Coloring and Style – **Color legend** = False

Inherit Style – **Plot** = Surface 1

– **Color** = True

– **Color and data range** = True

– **Deform scale factor** = False

– **Height scale factor** = False

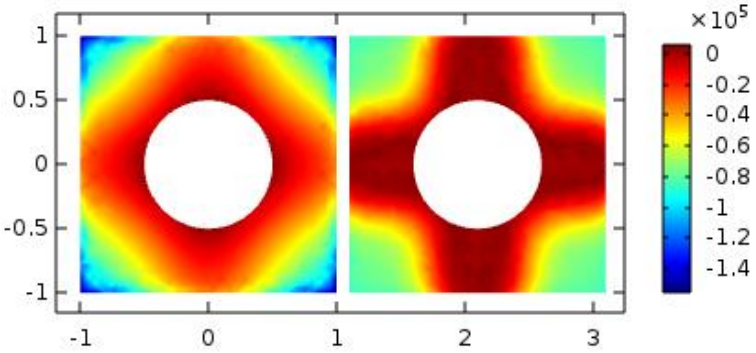
RMB > Deformation

2D Plot Group 6 – Surface 2 – Deformation 1:

Expression – **x component, y component** = 2.1,0 (*два поля*)

Scale – **Scale factor** = True, 1

LMB > Plot



Отже друге зображення зсунуто в графічній зоні праворуч на 2.1 одниці (метри).

На таких зображеннях корисно позначати положення точок максимального і мінімального значення. Для цього призначені графічні вузли типу Max/Min Surface. Графіки такого типу знаходять і будують на поверхні тіла/області точки мінімуму і максимуму виразу, який введено в полі Expression. Одночасно з цим створюється таблиця, в якій друкуються координати цих точок, і значення екстремумів в них.

Виконайте інструкції

2D Plot Group 6: RMB > More Plots - Max/Min Surface

2D Plot Group 6 – Max/Min Surface 1:

Data– **Data set** = Cut Plane 1

Expression – **Expression** = solid.sz

2D Plot Group 6: RMB > More Plots - Max/Min Surface

2D Plot Group 6 – Max/Min Surface 2:

Data– **Data set** = Cut Plane 2

Expression – **Expression** = solid.sz

Inherit Style – **Plot** = Surface 2

– **Color**= False

– **Deform scale factor** = True

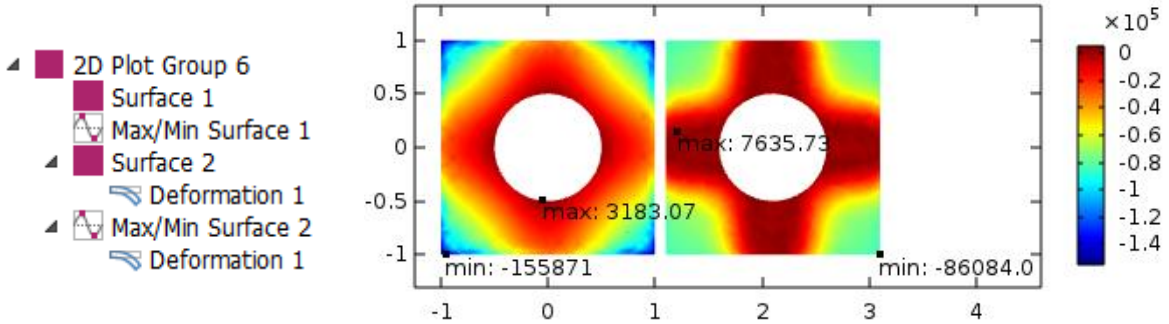
RMB > Deformation

2D Plot Group 6 – Max/Min Surface 2 – Deformation 1 :

Expression – **x component, y component** = 2.1,0 (*два поля*)

LMB > Plot

В результаті ми отримуємо графік напружень, на якому точками позначені максимальне і мінімальне (по перерізам) значення величини σ_z (наступний рисунок праворуч). Для зручності ми перемістили вузли Max/Min Surface в межах графічної групи так, як показано ліворуч.



Зверніть увагу на те, що графік **Max/Min Surface 2** успадковує від графіка **Surface 2** масштабний коефіцієнт, а не вектор зсуву, який треба задати явно. Зверніть також увагу на те, що в гілці **Tables** створюються дві таблиці з координатами точок екстремумів (по одній на кожний вузол **Max/Min Surface**).

Крім графіків, які позначають точки максимального і мінімального значення виразу по поверхні, можна будувати графіки максимуму і мінімуму по лінії або за об'ємом. Графік `Max/Min Line` знаходить і будує точки мінімуму і максимуму на лінії. Графік типу `Max/Min Volume` знаходить і малює точки максимуму і мінімуму в об’ємі тіла.

Перейдемо до розглядання лінійних графіків в тривимірних моделях. Їх сенс дещо відрізняється від одновимірних графіків функцій. Для пояснення створимо лінійний графік в новій тривимірній графічній групі.

`Results->`3D Plot Group

3D Plot Group 7:

`Data`– Data set = `Study 1/Solution 1`

`Plot Settings` – **Plot data set edges** = `False`

`3D Plot Group 6` **`->Add Plot`** – Line

3D Plot Group 7 – Line 1:

`Data`– Data set = `From parent`

`Expression` – **Expression** = `solid.disp`

`Coloring and Style` – **Line type** = `Tube`

`Coloring and Style` – **Tube radius expression** = `0.02`

`Coloring and Style` – **Radius scale factor** = `1`

RMB > Deformation

3D Plot Group 7 – Line 1 – Deformation 1:

`Expression` – **X component** = `u`

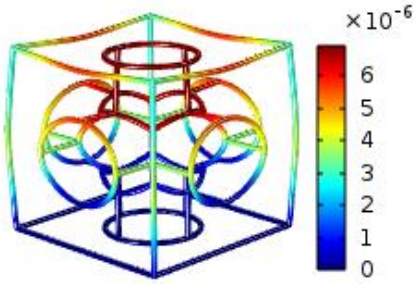
`Expression` – **Y component** = `v`

`Expression` – **Z component** = `w`

`Scale` – **Scale factor** = `True, 25000`

LMB > Plot

Клацніть ще по кнопці Show Grid, яка розташована вгорі графічного вікна, щоб прибрати відображення координатної сітки.



Отже `Line` графік «вибрав» з джерела даних `Study 1/Solution 1` всі одновимірні графічні елементи і розфарбував їх згідно виразу, зазначеному в полі **Expression**. В нашому випадку вираз `solid.disp` представляє модуль вектора переміщення. Зверніть увагу на те, що лінійний графік можна деформувати, додавши до нього дочірній вузол **Deformation**.

Для побудови лінійного графіка не на всіх ребрах тіла можна створити джерело даних типу `Edge 3D` і вказати для нього бажані ребра.

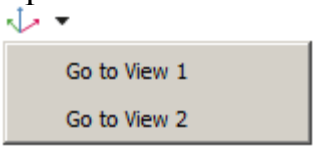
Results – Data Sets : RMB > Edge 3D

Data Sets - Edge 3D 1:

`Data` – **Data set** = `Study 1/Solution 1`

`Selection` – **Selection** =`1,14,77,84` (*вертикальні ребра куба*)

Зауваження. Якщо в графічному вікні при побудові набору даних **Edge 3D 1** використовується вид **View 2**, тобто фронтальні грані не відображаються, то переключіться на вид **View 1** за допомогою кнопки Go to Default 3D View (вона розташована вгорі ГВ), обравши бажаний вид з випадного списку.



Перейдемо до побудови графіка.

`Results->`3D Plot Group

3D Plot Group 8:

`Data`– **Data set** = `Edge 3D 1`

`3D Plot Group 8` **`-> Add Plot`** – Line

3D Plot Group 8 – Line 1:

`Data`– **Data set** = `From parent`

`Expression` – **Expression** = `solid.disp`

`Coloring and Style` – **Line type** = `Tube`

`Coloring and Style` – **Tube radius expression** = `0.02`

`Coloring and Style` – **Radius scale factor** = `1`

RMB > Deformation

3D Plot Group 8– Line 1 – Deformation 1:

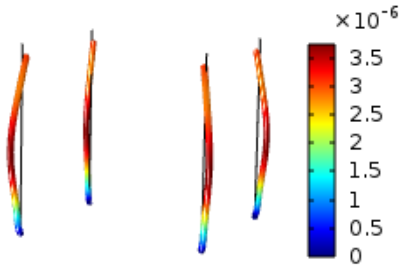
`Expression` – **X component** = `u`

`Expression` – **Y component** = `v`

`Expression` – **Z component** = `w`

`Scale` – **Scale factor** = `True, 50000`

LMB > Plot



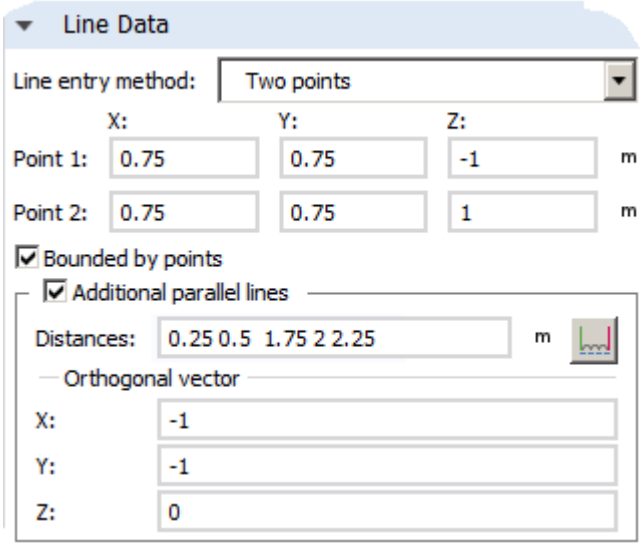
Для лінійного графіка можна створити окреме лінійне джерело даних. Для цього виконайте наступні інструкції.

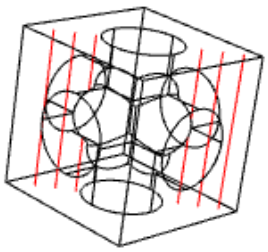
Results – Data Sets : RMB > Cut Line 3D

Data Sets - Cut Line 3D 1:

Data – **Data set** = Study 1/Solution 1

Далі заповніть ПВ створеного вузла **Cut Line 3D 1** згідно з наступним рисунком.





Тут показано, що лінійне джерело даних складається з базового відрізка, обмеженого двома точками із заданими координатами. Крім нього створюється ще 5 паралельних відрізків, розташованих на заданій відстані від базового відрізка в напрямку зазначеного вектора. Клацнувши по кнопці Plot, ви побудуєте зображення відрізків, з яких складається набір даних (попередній рисунок праворуч).

Тепер в кольорах зобразимо на заданих відрізках величину переміщення і додамо до лінійного графіка деформацію.

3D Plot Group 8 **-> Add Plot** – Line

3D Plot Group 8 – Line 2:

Data– **Data set** = Cut Line 3D 1

Expression – **Expression** = solid.disp

Coloring and Style – **Line type** = Tube

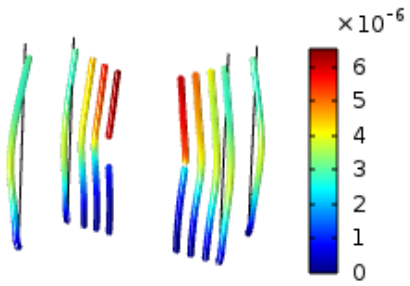
Coloring and Style – **Tube radius expression** = 0.02

Coloring and Style – **Radius scale factor** = 1

Inherit Style – **Plot** = Line 1

RMB > Deformation

LMB > Plot



Зверніть увагу на те, що лінійний графік **Line 2** успадковує всі стилі оформлення лінійного графіка **Line 1**, включно з забарвленням і масштабним коефіцієнтом.

Якщо одновимірне джерело даних складається з одного відрізка, то на ньому можна побудувати звичайний одновимірний графік, який можна розфарбувати схожим з лінійними графіками чином.

Results – Data Sets : RMB > Edge 3D

Data Sets - Edge 3D 2:

Data – **Data set** = Study 1/Solution 1

Selection – **Selection** =4 (*верхнє горизонтальне ребро куба*)

Results->1D Plot Group

1D Plot Group 9:

Data– **Data set** = Edge 3D 2

1D Plot Group 9 **-> Add Plot** – Line Graph

1D Plot Group 9 - Line Graph 1:

Data– **Data set** = From parent

y-Axis Data – **Expression** = solid.disp

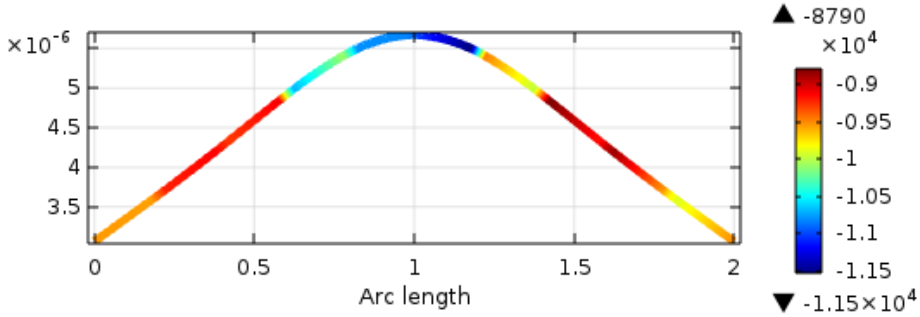
Coloring and Style – **Line, Width** = Solid, 5 (*два поля*)

RMB > Color Expression

1D Plot Group 9 - Line Graph 1 - Color Expression 1:

Expression – **Expression** = solid.sz

LMB > Plot



Отже крива зображає модуль вектора переміщення точок ребра, а колір – напруження σ_z в цих точках. Щоб задати колір лінії, який відповідає напруженню σ_z, до вузла **Line Graph 1** ми додали дочірній вузол **Color Expression 1**.

Покажемо, як можна побудувати зображення, яке наведено на початку цього прикладу. Воно ілюструє дію рівномірно розподіленої вертикальної сили на верхню грань тіла. Зображення самого тіла будується в тривимірній

графічній групі за допомогою Surface графіка з одиничним виразом, щоб колір всіх точок тіла був однаковий. Вектори на поверхні будуються як стрілковий графік векторного поля (0,0,-1). Arrow Surface графік (поверхнева векторна діаграма) прив'язує векторні поля до поверхонь (тобто початок або кінець стрілок розташований на поверхні). Щоб вектори були намальовані лише на верхній грані, для неї було створено окреме джерело даних Surface типу.

Results – Data Sets : RMB > Surface

Data Sets - Surface 1:

Data – **Data set** = Study 1/Solution 1

Parameterization – **x-and-y-axes** = XY-plane

Selection – **Selection** = 4 (*верхня грань куба*)

Results->3D Plot Group

3D Plot Group 10 **-> Add Plot** – Surface

3D Plot Group 10 – Surface 1:

Data– **Data set** = Study 1/Solution 1

Expression – **Expression** = 1

Coloring and Style – **Color table** = GrayScale

Coloring and Style – **Color legend** = False

3D Plot Group 10 **-> Add Plot** – Arrow Surface

3D Plot Group 10 – Arrow Surface 1:

Data– **Data set** = Surface 1

Expression – **x component** = 0

Expression – **y component** = 0

Expression – **z component** = -1

Coloring and Style – **Arrow base** = Head

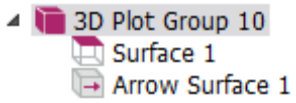
Coloring and Style – **Scale factor** = True, 0.6

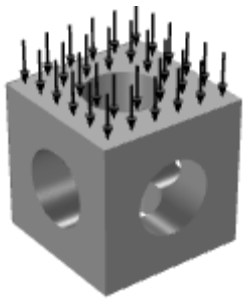
Coloring and Style – **Placement** = Uniform

Coloring and Style – **Number of arrows** = 30

Coloring and Style – **Color** = Black

LMB > Plot





Значення **Arrow base** = Head вказує на те, що на поверхні повинні бути розташовані кінці стрілок.

Звичайно, що в прикладних задачах замість діаграми сталого векторного поля (0,0,-1) використовуються розрахункові фізичні векторні поля.

Окрім поверхневих векторних діаграм можна будувати лінійні та об'ємні векторні діаграми.

Вправа 1. Побудуйте на поверхні тіла діаграму векторного поля переміщення (`u,v,w`).

Існує можливість створити копію основного джерела даних `Study 1/Solution 1`, в якому використати не всі елементи геометрії. При створенні вузла-копії в його підвузлі **Selection** можна обрати рівень геометричної сутності (level), і потім вказати потрібні елементи геометрії. Це схоже на приховування геометричних елементів у вузлах `View`. Наприклад, зобразимо тільки зовнішню поверхню нашого тіла і на ній контурний графік величини переміщення.

Data Sets - Study 1/Solution 1 : RMB > Duplicate
Data Sets - Study 1/Solution 1 (2) : RMB > Selection
Data Sets - Study 1/Solution 1 (2) - Selection:

`Geometric Entity Selection` – **Geometric entity level** = `Boundary`
`Geometric Entity Selection` – **Selection** = `1,2,3,4,9,30` (*зовнішні поверхні тіла*)

Після створення копії джерела даних для неї (копії) ми додали підвузол **Selection**, який дозволяє обрати для цієї копії рівень геометрії, який буде використовуватися. В нашому випадку ми обрали зовнішні границі тіла.

Зверніть також увагу на те, що після створення копії ДД автоматично перейменовується вузол `Study 1/Solution 1`, який тепер всюди в моделі буде називатися `Study 1/Solution 1(1)`.

Новий набір даних `Study 1/Solution 1(2)` можна використовувати при побудові графіків.

`Results->`3D Plot Group

3D Plot Group 11:

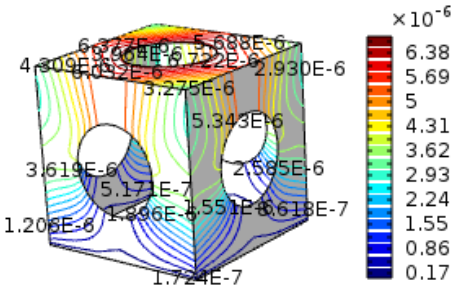
`Data`– **Data set** = `Study 1/Solution 1(2)`
`3D Plot Group 11` **`->Add Plot`** – Surface

3D Plot Group 11 – Surface 1:

`Data`– **Data set** = `From parent`
`Expression` – **Expression** = `1`
`Coloring and Style` – **Color table** = `GrayScale`
`Coloring and Style` – **Color legend** = `False`
`3D Plot Group 11` **`->Add Plot`** – Contour

3D Plot Group 11 – Contour 1:

`Data`– **Data set** = `From parent`
`Expression` – **Expression** = `solid.disp`
`Coloring and Style` – **Level labels** = `True`
LMB > Plot



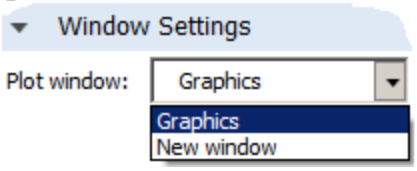
Використання для контурного графіка прапорця **Level labels** = `True` встановлює числові значення на контурних лініях.

В копії ДД можна обрати **Geometric entity level** = `Edge`. Тоді розв'язок буде дублюватися на кромках/ребрах тіла. Якщо при цьому задати

`Geometric Entity Selection` – **Selection** = `All Edges`

то нове ДД можна використати в лінійному графіку для креслення ребер геометрії (**Expression**=`1`, **Color**=`Black`). Використання вузла `Deformation` додасть до контуру геометрії деформацію.

При побудові графіків завжди є можливість відображати їх в окремому вікні. Для цього в ПВ графічних груп є розділ **`Window Settings`**, в якому виконується потрібний вибір.



Звичайно, що вікна графіків можна перетягати, розташовуючи кілька вікон поруч.

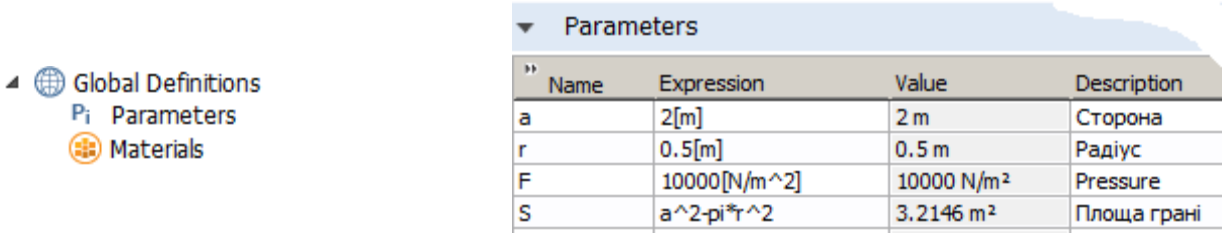
Збережіть файл, наприклад, з ім'ям `cube_with_holes.mph`. Ми продовжимо роботу з ним в наступному параграфі.

1.5.4. Параметри, змінні, зонди (probes).

Параметри. В багатьох виразах Comsol Multiphysics застосовуються параметри (Parameters) – іменовані числові константи, які можна використовувати в будь-яких місцях моделі замість чисел. Імена та значення параметрів задаються в таблиці параметрів у ПВ вузла **Parameters** глобальної гілки визначень **Global Definitions**. Сам вузол параметрів створюється командою

`Home` **`-> Definitions`** – Parameters

або аналогічною командою з випадного меню.



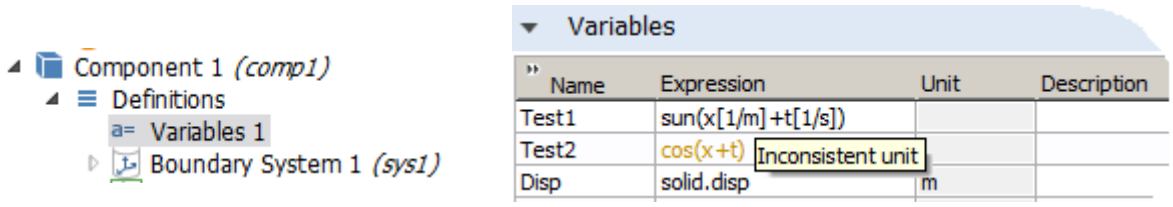
В таблиці параметрів в колонці **Name** вводять імена, а в колонці **Expression** – значення параметрів. Поля в стовпці **Value** заповнюються автоматично. В колонці **Description** вводяться необов'язкові коментарі. Вони іноді

з'являються в позначках на графіках, в яких не передбачено використання кирилиці. Тому на попередньому рисунку змінну тиску `F` ми позначили «Pressure».

Змінні. Крім параметрів в системі використовуються змінні (Variables). Змінні – це назви просторових координат (наприклад, x, y, z), ідентифікатор часу `t`, імена шуканих величин, наприклад, T - температура. В прикладних режимах існує багато спеціальних змінних. Користувач може створювати власні змінні з інших, використовуючи припустимі операції. Змінні роблять модель простішою для розуміння шляхом введення коротких та наочних імен для складних виразів.

Змінні користувача задаються в таблицях вузлів **Variables**. Ці вузли створюються в глобальній гілці визначень **Global Definitions**, або локально в межах компонента. Наприклад, вузол локальних змінних можна створити командою

`Home` **`->Definitions`** – Variables – Local Variables



Вирази для обчислювання змінних можуть містити числа, параметри, математичні або фізичні константи, інші змінні, функції від змінних, знаки математичних операцій, та оператори, наприклад, диференціювання, інтегрування або обчислювання середнього значення. Змінні можуть також залежати від шуканих в задачі величин та їх похідних.

Більшість змінних, таких як координати, матеріальні властивості чи шукані величини, залежать від розрахункової області і являють «полеві» змінні, тобто вони залежать від просторових координат та часу. Наприклад, `solid.rho` – вбудована змінна густини в режимі `Solid Mechanics`, представляє її як функцію $\rho(x, y, z, t)$. Ви можете побудувати графік цієї функції, але не зможете використати в ЗДР, навіть якщо вона є сталою.

Змінні стають доступними після виконання команди Compute. При внесенні модифікацій в таблицю змінних, ви повинні виконати команду Update Solution, щоб прийняти додані зміни до уваги, або заново переобчислити модель. При цьому слід мати на увазі, що до виконання команди Compute перший раз, вирази в стовпці **`Expression`** майже завжди будуть виділені помаранчевим кольором тому, що система ще нічого не знає про імена змінних. Після виконання команди Compute змінні стають відомими, і система може проконтролювати вираз і його розмірність.

Параметри і змінні глобальної (зовнішньої) гілки визначень **Global Definitions** можна використовувати в будь-яких місцях моделі. Елементи локальної гілки **Component – Definitions** призначені для використання в межах компонента, але можуть застосовуватися і поза компонентом за допомогою

точкової нотації. Отже параметри завжди глобальні, а змінні можуть бути глобальними і локальними в залежності від того, де ви їх визначили.

На відміну від параметрів змінні не можна використовувати в гілках **Geometry** та **Mesh**, а в гілці **Study** їх можна використовувати лише обмежено.

В Comsol Multiphysics існує ще один специфічний тип змінних, які називаються зондами (Probes). Вони створюються в локальних вузлах **Definitions** компонентів і являють собою скаляри (сталі, функції часу або параметрів, але не координат). Зазвичай вони характеризують якусь скалярну властивість компонента, наприклад, середнє значення розрахункової величини або значення величини в точці. Якщо задача динамічна, то ці змінні, можуть залежати від часу. Їх можна використовувати в коефіцієнтах рівнянь, граничних умовах, в умовах контролю або зупинки обчислень (в гілці **Study**). Зондові змінні (probe variables) використовуються також після завершення обчислень для аналізу результатів.

Продемонструємо використання змінних, продовживши роботу з файлом `cube_with_holes.mph`, який було створено в попередньому параграфі. Відкрийте цей файл.

Припустимо, що нам потрібно обчислити загальну силу, яку чинить тіло на основу. Вона дорівнює вазі тіла плюс зовнішня сила, яка діє на тіло зверху. Для виконання такого обчислення в COMSOL Multiphysics існує декілька інструментів. Найпростіший з них – це використання в гілці **Derived Values** вузла типу **Volume Integration**. Але цей спосіб доступний лише на етапі постобробки результатів (в гілці **Results**). Якби вам було потрібно використати значення об’єму тіла або загальну силу, яку воно чинить на основу, в іншому компоненті або всередині математичного виразу поточного компоненту, то такий підхід не спрацює. Загальний спосіб полягає в використанні глобальних або локальних змінних і зондів. Вони можуть зберігати значення об’єму тіла, густину його речовини або якесь інше значення, що характеризує тіло загалом. Якщо зонд, локальна або глобальна змінна створені, то їх можна використовувати в будь-якому місці моделі: при побудові графіків, в виразах поточного компоненту, або в іншому компоненті моделі. При цьому для зондів і локальних змінних бажано (а часом обов’язково) використовувати точкову нотацію.

Глобальні змінні корисні переважно для позначення величин, які не залежать від точок обчислювання. Наприклад, вони можуть позначати змінні, які обчислюються в компоненті шляхом розв’язання ЗДР або алгебраїчного рівняння, і значення яких не залежить від геометрії моделі. Якщо змінна характеризує тіло, то ймовірно, її слід створювати як локальну до компонента, в якому створено геометрію тіла. Якщо змінна характеризує не тільки тіло, а залежить від зовнішніх факторів, то її логічно визначати як глобальну змінну.

Для обчислення змінної, яка представляє об’єм чи масу тіла потрібно виконувати інтегрування, але таке, яке може бути викликано з будь якого місця моделі. Операція інтегрування, яка виконується в межах компонента і може визиватися в будь якому місці моделі відноситься до групи операторів зв’язку (Component Couplings). Такі оператори можуть використовуватися для передачі

значень між моделями різних фізичних процесів в межах одного компонента або між компонентами, а також на етапі обробки результатів. Характер значень, що передаються, може бути різним. Зокрема, це можуть бути деякі інтегральні характеристики моделі, які обчислюються оператором зв’язку `Integration`.

Оператори зв'язку (Coupling Operators) визначаються в гілці **Definitions** компонента, з якого вибираються дані. Тут оператори тільки визначаються, але не обчислюються. Однак після створення їх можна використовувати на будь-якому етапі моделювання в будь-якому місці поточного або іншого компонента. Для створення операторів зв'язку використовується кнопка - список Component Coupling панелі інструментів `Definitions` або однойменна команда з випадного меню локального вузла **Definitions**.

Припустимо, що ми бажаємо обчислити об'єм тіла за допомогою інтеграла $\iiint\limits_V 1\,dx\,dy\,dz$. Для цього зручно використовувати оператор зв'язку `Integration`. Подивимося, як це робиться.

`Definitions` -> **`Component Coupling`** – Integration

Активуйте ПВ створеного вузла **Component 1 (comp1) - Definitions - Integration 1 (intop1)**. Ім'я оператора інтегрування `intop1` вказується в полі **Operator name**. Далі введіть:

Label: = `Інтеграл за об'ємом`

`Source Selection` – **Geometric entity level** = `Domain`

Остання інструкція вказує тип області інтегрування. В асоційованому графічному вікні клацніть по області тіла, і його індикатор 1 з'явиться в списку **Selection**, вказуючи, тим самим, конкретну область.

Обновіть розв’язок, клацнувши по кнопці Update Solution в ПВ вузла **Study 1**.

Тепер оператор `intop1` створений і доступний. Продемонструємо його використання в нашій моделі.

`Results` -> **`Derived Values`** – Global Evaluation

Results - Derived Values - Global Evaluation 1:

`Expression` – **Expression** = `comp1.intop1(1)`

LMB > Evaluate – New Table

В моделі з’явиться таблиця **Table 1** і в ній комірка з результатом обчислення. Оскільки інтегрування одиниці виконувалося по всьому тілу, то результатом є об’єм тіла.

comp1.intop1(1) (m^3)
4.7019

Як бачите, на етапі постобробки, використання оператора зв’язку `Integration` ідентично використанню залежної величини `Volume Integration` (тобто вузла **Derived Values – Integration – Volume Integration**).

Зауваження. В списку **Geometric entity level** вузла **Definitions - Інтеграл за об'ємом (intop1)** в якості типу області інтегрування можна вибрати значення `Boundary` (поверхня), `Edge` (ребро) і `Point` (точка). При виборі рівня геометрії

Edge інтегрування буде виконуватися по зазначеній кривій. Якщо в якості рівня інтегрування використати точку, то оператор Integration поверне значення виразу в точці. Якщо на етапі створення оператора інтегрування intop(expr) вибрати кілька точок, то буде обчислюватися сума значень виразу expr в цих точках.

Оператор зв'язку Integration можна використовувати не тільки для обчислення значень на етапі обробки результатів. Його можна використовувати на будь-якому етапі моделювання, в тому числі при створенні глобальних і локальних змінних. Наприклад, створимо локальну змінну, яка являє об'єм тіла.

Якщо в вашій моделі ще немає вузла локальних змінних **Component 1 – Definitions – Variables 1**, то створіть його командою

Home **-> Definitions** – Variables – Local Variables

Якщо такий вузол вже є, то очистіть рядки його таблиці змінних за допомогою кнопки Delete, яка розташована під таблицею.

В ПВ вузла **Component 1 – Definitions – Variables 1** в першому рядку таблиці змінних введіть:

Variables – **Name** = Vol1
– **Expression** = comp1.intop1(1)

Study 1: RMB > Update Solution

Тепер перевіримо працездатність змінної Vol1, надрукувавши її значення.

Results -> **Derived Values** – Global Evaluation

Results - Derived Values - Global Evaluation 2:

Expression – **Expression** = comp1.Vol1

LMB > Evaluate – Table 1

intop1(1) (m^3)	Vol1 (m^3)
4.7019	4.7019

Один і той же оператор intop1 можна використовувати для інтегрування різних виразів. Наприклад, intop1(густина) обчислить масу тіла.

Створимо локальну змінну Mass1, що містить значення маси тіла. Для цього у другому рядку таблиці вузла **Component 1 – Definitions – Variables 1** введіть:

Variables – **Name** = Mass1
– **Expression** = comp1.intop1(comp1.mat1.def.rho)

Study 1: RMB > Update Solution

Зверніть увагу на те, як використовується ім'я, що представляє густину тіла. Для доступу до властивості rho матеріалу mat1 потрібно використовувати проміжне ім'я групи властивостей def.

Зауваження. Для посилання на властивості матеріалу потрібно використовувати його ім'я (наприклад, mat1), і дескриптор/тег групи базових властивостей, який зазвичай називається def. Наприклад, для використання густини речовини можна використовувати позначення mat1.def.rho (або повне ім'я root.comp1.mat1.def.rho). Для доступу до цієї величини в межах

«фізики» перед її ім'ям використовуйте ідентифікатор «фізики», наприклад, `solid.rho`.

Надрукуємо масу.

`Results` -> **`Derived Values`** – Global Evaluation

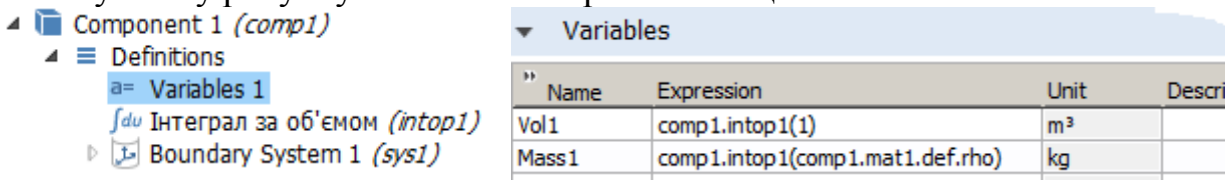
Results - Derived Values - Global Evaluation 3:

`Expression` – **Expression** =`comp1.Mass1`

LMB > Evaluate – Table 1

comp1.intop1(1) (m^3)	comp1.Vol1 (m^3)	comp1.Mass1 (kg)
4.7019	4.7019	10814

На наступному рисунку показана створена таблиця локальних змінних.



Тепер створимо глобальну змінну, що зображає загальну силу `Force`, яку чинить тіло на основу. Вона дорівнює вазі тіла плюс зовнішня сила, яка діє на тіло зверху

Глобальні змінні створюються аналогічно до локальних. Оскільки змінні `Vol1` і `Mass1` відносяться до тіла, створеного в компоненті, то ми їх визначили як локальні. Змінна `Force` характеризує не тільки тіло, а залежить від зовнішнього фактора (тиску на верхню грань), і її логічно визначити як глобальну змінну.

Створимо таблицю глобальних змінних командою

`Home` -> **`Definitions`** – Variables – Global Variables

В ПВ створеного вузла **Global Definitions - Variables 2** в першому рядку таблиці змінних введіть:

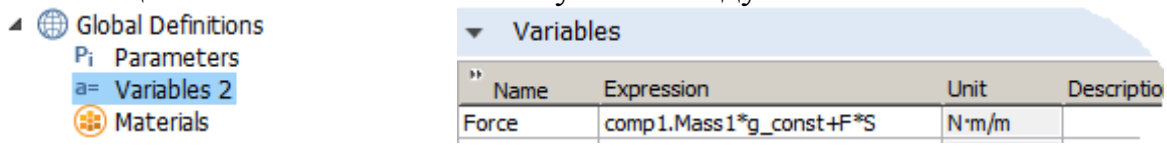
`Variables` - **Name** = `Force`

- **Expression** = `comp1.Mass1*g_const+F*S`

Study 1: RMB > Update Solution

Нагадаємо, що стала `g_const` позначає прискорення вільного падіння.

Таблиця глобальних змінних набуває вигляду



Надрукуємо значення глобальної змінної `Force`.

`Results` -> **`Derived Values`** – Global Evaluation

Results - Derived Values - Global Evaluation 4:

`Expression` – **Expression** = `Force`

LMB > Evaluate – Table 1

comp1.intop1(1) (m^3)	comp1.Vol1 (m^3)	comp1.Mass1 (kg)	Force (N)
4.7019	4.7019	10814	1.3820E5

Окрім змісту, треба враховувати зону дії змінних. Іноді корисно обмежити контекст змінної до певної частини геометрії, в якій вона доступна, наприклад,

до певної частини тіла, до поверхні, або лише до деяких ребер і навіть окремих точок.

Припустимо, що нам потрібно побудувати графік напруження σ_z вздовж квадратного контура основи тіла. Це можна зробити декількома способами. Один з них полягає в побудові локальної змінної, яка визначена лише на цьому контурі.

Створимо вузол локальних змінних, зона дії яких буде обмежена зазначеним контуром.

Home **-> Definitions** – Variables – Local Variables

Component 1 – Definitions – Variables 3:

Geometric Entity Selection – **Geometric entity level** = Edge

– **Selection** = Manual; 2,3,15,78 (*ребра основи*)

В першому рядку таблиці змінних введіть:

Variables – **Name** = Sz

– **Expression** = solid.sz

Study 1: RMB > Update Solution

Нам знадобилося побудувати новий вузол **Variables 3** тому, що його змінні визначені на іншому рівні геометрії (на ребрах), в протилежність від змінних вузла **Variables 1**, які визначені по всій моделі.

Побудуємо графік змінної Sz над заданим контуром.

Results->3D Plot Group

3D Plot Group 12:

Data– **Data set** = Study 1/Solution 1(1)

Plot Settings – **Plot data set edges** = False

3D Plot Group 12 **-> Add Plot** – Line

3D Plot Group 12 – Line 1:

Data– **Data set** = From parent

Expression – **Expression** = comp1.Sz

Coloring and Style – **Line type** = Tube

– **Tube radius expression** = 0.02

– **Radius scale factor** = 1

Quality – **Recover** = Within domains

RMB > Deformation

3D Plot Group 12 – Line 1 – Deformation 1:

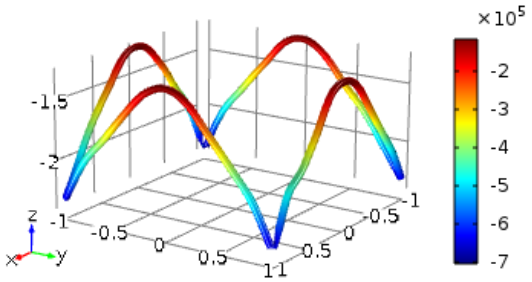
Expression – **X component** = 0

– **Y component** = 0

– **Z component** = comp1.Sz

Scale – **Scale factor** = True,7e-6

LMB > Plot



Зверніть увагу, що для поліпшення графіка ми включили опцію відновлення похідних **Recover** = `Within domains`.

Для наочності побудуємо цей графік, «розгорнувши» контур в пряму лінію.

`Results->`1D Plot Group

1D Plot Group 13:

`Data`– Data set = `Study 1/Solution 1(1)`

`1D Plot Group 13` **`->Add Plot`** – Line Graph

1D Plot Group 13 – Line Graph 1:

`Data`– Data set = `From parent`

`Selection` – **Selection** = `Manual; 2,3,15,78` (*ребра основи*)

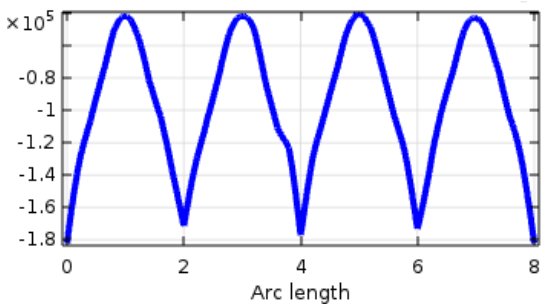
`y-Axis Data` – **Expression** = `comp1.Sz`

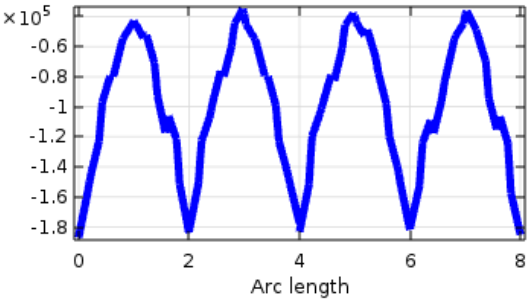
`x-Axis Data` – **Parameter** = `Arc length`

`Coloring and Style` – **Line, Width** = `Solid,3` (*два поля*)

`Quality` – **Recover** = `Within domains`

LMB > Plot (*наступний графік ліворуч*)





Якщо вимкнути опцію відновлення похідних **`Quality`** – **Recover** = `Off`, то графік буде виглядати так, як зображено на попередньому рисунку праворуч.

Існує також спосіб об’єднання різнойменних змінних, визначених в різних частинах геометрії, під єдиним ім’ям. Для цього слід створити кілька вузлів `Variables`, зону дії яких обмежити потрібними частинами геометрії. Потім слід задати в цих вузлах змінну з однаковим ім’ям і відповідним виразом для обчислення. Звернення до такого імені, наприклад, при побудові графіка, буде автоматично використовувати вираз на відповідній частині геометрії.

Параметричний аналіз. Припустимо тепер, що нам потрібно дослідити модель в залежності від якогось параметра. Для цього можна було б змінити значення параметра, обчислити модель, запам’ятати розв’язок, змінити параметр, знову обчислити модель, знову запам’ятати розв’язок і т.д. Це клопітно, тому в системі існують відповідні інструменти аналізу залежності розв’язків від параметрів. Звичайно такі параметри повинні бути створені у вузлі глобальних

параметрів (в COMSOL Multiphysics інших параметрів не буває). Потім у вузлі **Study** встановлюються ознаки (або створюється вузол, що містить ознаки) вмикання параметричного аналізу.

Якщо параметр керує геометрією моделі (репрезентує довжину, товщину, радіус, тощо), то в гілці **Study** слід створити підвузол **Parametric Sweep**, і в його ПВ вказати параметр і список значень, що потрібно проаналізувати. Якщо параметр використовується в початкових або граничних умовах, являє матеріальний коефіцієнт або коефіцієнт рівняння (тобто не змінює геометрію), то краще використовувати `Auxiliary Sweep` (допоміжний) аналіз, який вмикається встановленням відповідного прапорця в ПВ вузла **Study – Step 1: Stationary**.

Виконаємо в нашій моделі допоміжний параметричний аналіз (`Auxiliary Sweep`) по параметру `F` – тиску на верхню грань. Для цього виконайте інструкції:

Study 1 – Step 1: Stationary:

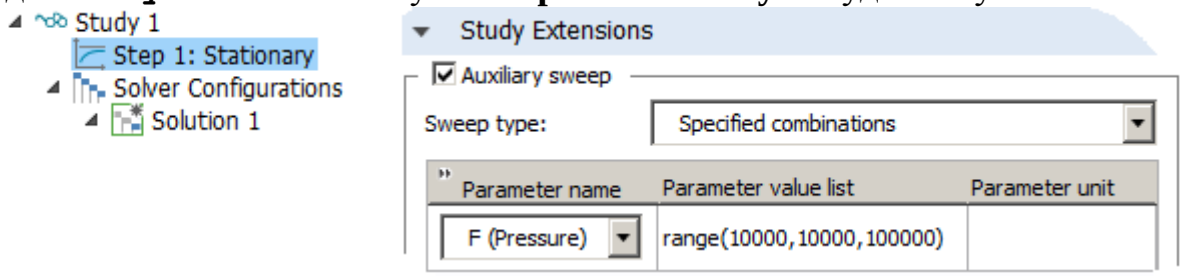
`Study Extensions` – **Auxiliary sweep** = `True`

В ПВ вузла **Step 1: Stationary** в розділі **`Study Extensions`** активується таблиця вибору параметрів. Клацніть по таблиці правою кнопкою миші і з випадного меню виберіть команду Add. Зі списку, який з'явиться в стовпці **Parameter name**, виберіть параметр `F`. У другому стовпці таблиці задайте діапазон зміни цього параметра `range(10000,10000,100000)`. Ці дії можна записати у вигляді наступних інструкцій:

`Study Extensions` – **Parameter name** = `F(Pressure)`

– **Parameter value list** = `range(10000,10000,100000)`

Розділ **`Study Extensions`** вузла **Step 1: Stationary** набуде наступного вигляду.



Після цього виконайте обчислення моделі.

`Study`-> Compute

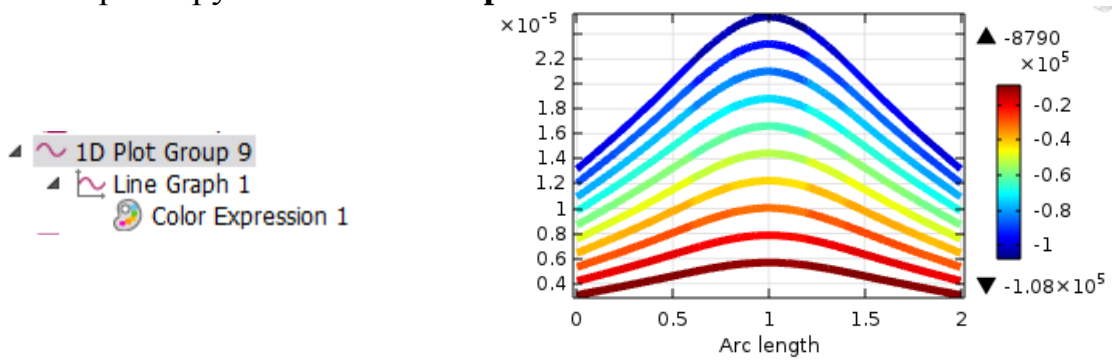
В панелях властивостей майже всіх графічних груп і деяких графіків в розділі **`Data`** з'явиться список **Parameter value (F)** зі значенням параметра `F`, для якого побудовано графік. Ви можете обрати будь-яке значення з діапазону `10000,20000,...,100000`, який було задано в таблиці.

Зверніть увагу, що у вузлі **Stress (solid) - Surface 2 - Deformation 1** потрібно встановити масштабний коефіцієнт, бо незважаючи на те, що він повинен успадковуватися з вузла **Surface 1**, це не відбувається.

Stress (solid) - Surface 2 - Deformation 1:

`Scale` – **Scale factor** = `True, 25000`

В одновимірних групах будуються графіки для всіх 10 значень параметра. Ось, наприклад, як будуть виглядати графіки переміщення точок ребра в одновимірній групі **1D Plot Group 9**.



Таким же чином перебудовуються графіки в групі **1D Plot Group 13**.

Пояснимо ще раз різницю між параметричними аналізами `Parametric Sweep` і `Auxiliary Sweep`. Останній зручно використовувати, коли при параметричному аналізі не виконуються геометричні зміни, тобто змінюються параметри зовнішнього впливу, граничні або початкові значення, коефіцієнти рівнянь, тощо. При `Parametric Sweep` аналізі можна змінювати параметри, які керують геометрією (тобто товщину, довжину, діаметр, тощо). Поведінка COMSOL залежить від того, який параметричний аналіз обрано. При `Parametric Sweep` аналізі зміни вносяться по всій моделі: в геометрію, СкЕ сітку і «фізику». Потім виконується обчислення моделі, параметр набуває нового значення і все повторюється. Тобто для кожного нового значення параметра дані у вузлах **Geometry** , **Mesh** і «**фізика**» оновлюються, і повторно виконуються обчислення.

При `Auxiliary Sweep` аналізі вважається, що параметр змінюється неперервно. Зміни вносяться лише в «фізику», і потім виконується обчислення. Результат, отриманий на попередньому кроці розглядається як початкове значення для нової ітерації при новому значенні параметра. У випадку `Auxiliary Sweep` солвер не виконує зміни у вузлах **Geometry** і **Mesh**. Це зменшує час обчислення і покращує точність.

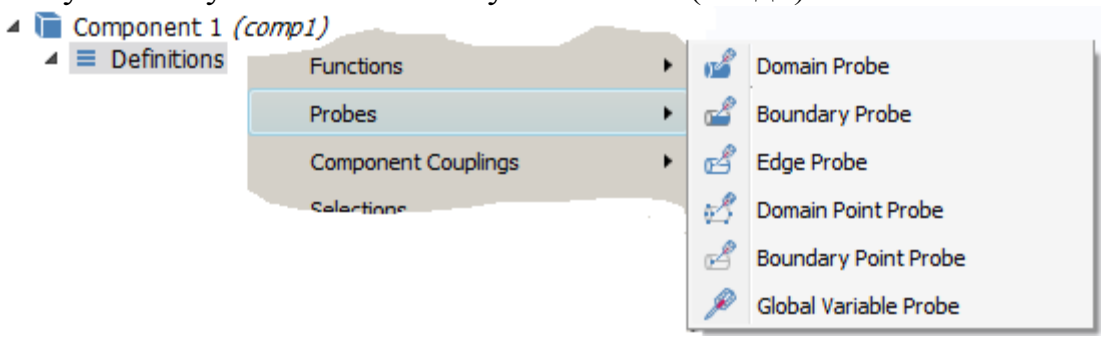
Зауваження. Параметричний аналіз можна виконувати відразу по кільком параметрам.

Зонди (probes). Уявіть собі експеримент, в якому використовується деякий давач (або сенсор), що контролює якийсь фізичний параметр в певній точці. При комп'ютерному моделюванні фізичних явищ бажано мати схожий інструмент. В COMSOL Multiphysics він називається зонд (probe). В динамічних задачах зонди можуть використовуватися для контролювання полевих змінних в точках під час обчислювання, але мають більш гнучкі можливості порівняно з давачами/сенсорами, які контролюють полеві змінні лише в точках.

В COMSOL Multiphysics зонд – це скалярна змінна, яка не є функцією просторових координат. Як і фізичний давач, підключений до зовнішніх інтерфейсів виведення, зонд «друкує» дані в таблицю і по ній будує графік відповідної змінної (в статичних задачах в залежності від параметра, а в

динамічних – від часу). Крім того, для свого функціювання він автоматично «створює» власні джерела даних.

Список доступних типів зондів можна побачити, якщо розкрити у випадному меню вузла **Definitions** пункт Probes (зонди).



Перші 3 типи зондів схожі за дією. Це «просторові» зонди (domain probes), «поверхневі» зонди (boundary probes) і «реберні» зонди (edge probes). Вони дозволяють відслідковувати середнє, мінімальне, максимальне або інтегральне значення «полевої» величини по області, поверхні або вздовж ребра.

В розглядуваній моделі створимо `Boundary` зонд, що представляє середнє напруження σ_z на нижній грані.

Component 1 (comp1) – Definitions:

RMB: Probes – Boundary Probe

В ПВ створеного вузла **Component 1 – Definitions - Boundary Probe 1 (bnd1)** спорожніть секцію вибору кнопкою Clear Solution і додайте індикатор 3 нижньої грані тіла. Потім введіть

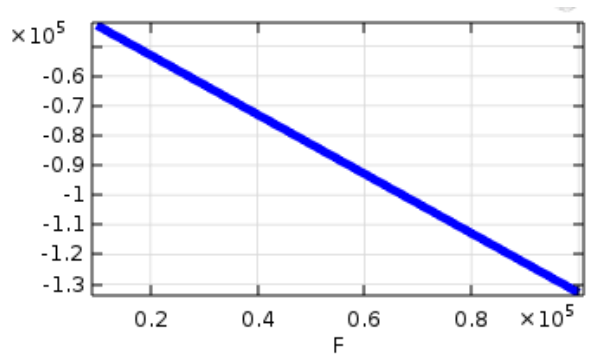
`Probe Type` – **Type** = `Average`

`Expression` – **Expression** = `solid.sz`

Study 1: RMB > Compute

Після виконання команди Compute в гілці **Data Sets** створюються два джерела даних: **Probe Solution 3** і **Boundary Probe 1**. Перше є копією **Study 1/Solution 1 (1)**, а друге – «звуженою» копією з поверхневим рівнем геометрії. В гілці **Derived Values** створюється вузол **Boundary Probe 1** типу `Point Evaluation`. В його списку **Data set** обрано «звужений» однойменний набір даних **Boundary Probe 1**, а поле **Expression** містить вираз `solid.sz`, взятий з такого ж поля зонда. Автоматичне обчислення вузла **Boundary Probe 1** створює таблицю **Probe Table 2** і друкує в ній середні значення напруження `solid.sz` при різних значеннях зовнішнього навантаження (параметра) `F` (наступний рисунок ліворуч).

F	Stress tensor, z co	, Boundary Probe 1
10000	-42991	
20000	-52980	
30000	-62970	
40000	-72959	
50000	-82949	
60000	-92938	
70000	-1.0293E5	
80000	-1.1292E5	
90000	-1.2291E5	
1.0000E5	-1.3290E5	



Також автоматично за цією таблицею створюється одновимірна графічна група **Probe Plot Group 14** і в ній будується графік (типу `Table Graph`) залежності середнього тиску від величини зовнішнього навантаження `F` (попередній рисунок праворуч).

В системі існують 2 типи точкових зондів: `domain point` і `boundary point`, які схожі на давачі, що контролюють фізичний параметр в певній точці. Ці зонди являють змінні, які містять значення «полевих» величин в точках, розташованих в областях або на поверхнях.

Створимо в нашій моделі точковий зонд, який відслідковує вертикальний тиск в нижній вершині куба.

Component 1 (comp1) – Definitions:

RMB: Probes – Boundary Point Probe

В ПВ створеного вузла **Component 1 – Definitions - Boundary Point Probe 1** в секцію вибору додайте індикатор 3 нижньої грані тіла (тобто точка буде обиратися на цій грані). Потім введіть

Definitions - Boundary Point Probe 1:

`Point Selection` – **x,y,z** = `-1,-1,-1` (*три поля*)

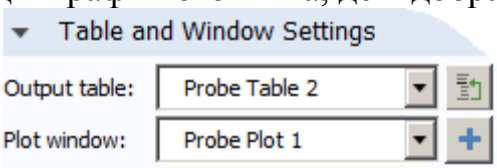
Definitions - Boundary Point Probe 1 – Point Probe Expression 1 (ppb1):

`Expression` – **Expression** = `solid.sz`

Study 1: RMB > Compute

Після виконання команди Compute в гілці **Data Sets** створюється два джерела даних: **Probe Solution 4** (типу `Solution`) і **Boundary Point Probe 1** (типу `Cut Point 3D`) з підвузлом **Point Probe Expression 1(ppb1)**. В гілці **Derived Values** створюється вузол **Point Probe Expression 1** типу `Point Evaluation`. Цей вузол обчислюється автоматично і результат розміщується в попередній таблиці **Probe Table 2**. Також автоматично за цією таблицею в існуючій графічній групі **Probe Plot Group 14** перебудовується графік, на який додається крива, що зображає залежність тиску σ_z в вершині основи від величини зовнішнього навантаження `F` (зелена крива на графіку на наступній сторінці).

Перший створений зонд моделі побудував в гілці **Results – Tables** «зондову» таблицю для відображення чисельних результатів. Одночасно з цим була створена одновимірна графічна група з `Probe Table Graph` підвузлом, в якому зображені зондові дані у вигляді лінійного графіка. Всі наступні зонди за замовчуванням друкують свої значення в цю спільну таблицю (в нашому прикладі в **Probe Table 2**), і всі «зондові» криві кресляться на спільному графіку (в нашому прикладі на графіку **Probe Table Graph 1** графічної групи **Probe Plot Group 14**). Це визначається налаштуваннями в розділ **`Table and Window Settings`** вікон властивостей зондів. В ньому можна налаштувати «адресу» вихідної таблиці і графічного вікна, де відображаються зондові змінні.



Зауваження. В точкових зондах можна створювати необмежену кількість підвузлів `Point Probe Expression`, в кожному з яких можна обчислювати власний вираз (в тій самій точці). Безпосередньо ці підвузли містять поле, в якому задається ім'я зондової змінної, і саме вони являють зондові змінні, а не батьківський вузол. За замовчування у вузлах `Domain Point Probe` і `Boundary Point Probe` завжди створюється один підвузол `Point Probe Expression`.

Для тренування створимо зонд типу `Global Variable Probe`. Такі зонди не використовують геометрію і відносяться до всього компонента. Їм надаються власні імена, і в полі **Expression** вказується вираз, який це ім'я буде заміняти. Отже виконайте наступні інструкції.

Component 1 (comp1) – Definitions:

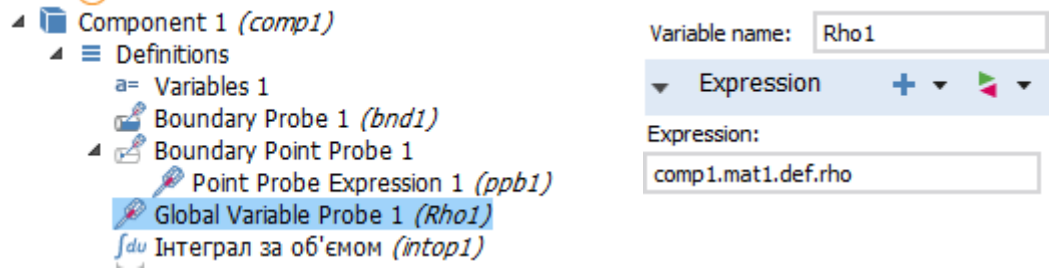
RMB: Probes – Global Variable Probe

Component 1 (comp1) – Definitions – Global Variable Probe 1:

Variable Name = `Rho1`

`Expression` – **Expression** = `comp1.mat1.def.rho`

Study 1: RMB > Compute



Так само, як і для попередніх зондів, після виконання команди Compute, створюються кілька вузлів ДМ, але нових джерел даних не створюється. Інформація про зонд зберігається в існуючому вузлі ДД **Probe Solution 3**. В гілці **Derived Values** створюється вузол **Global Variable Probe 1**. Він автоматично обчислюється і результат розміщується в попередній таблиці **Probe Table 2**. Також автоматично за цією таблицею в графічній групі **Probe Plot Group 14** перебудовується графік, на який додається крива, що зображає залежність густини від величини зовнішнього навантаження `F`. Звичайно залежності немає і в таблиці в відповідній колонці друкуються однакові значення.

Для зручності скоротимо назви стовпців таблиці **Probe Table 2**. Активуйте ПВ цього вузла і відредагуйте таблицю в розділі **`Column Headers`** так, як зображено на наступному рисунку ліворуч. Натисніть кнопку Update і отримайте таблицю, яка зображена на наступному рисунку праворуч.

Column Headers

Column	Header
1	F
2	Boundary Probe 1
3	Point Probe 1
4	Global Probe 1

F	Boundary Probe 1	Point Probe 1	Global Probe 1
10000	-42991	-1.8501E5	2300.0
20000	-52980	-2.4389E5	2300.0
30000	-62970	-3.0278E5	2300.0
40000	-72959	-3.6166E5	2300.0
50000	-82949	-4.2055E5	2300.0
60000	-92938	-4.7944E5	2300.0
70000	-1.0293E5	-5.3832E5	2300.0
80000	-1.1292E5	-5.9721E5	2300.0
90000	-1.2291E5	-6.5609E5	2300.0
1.0000E5	-1.3290E5	-7.1498E5	2300.0

Разом з таблицею оновляться графіки в групі **Probe Plot Group 14**. Щоб в групі відобразити легенду, виконайте інструкції

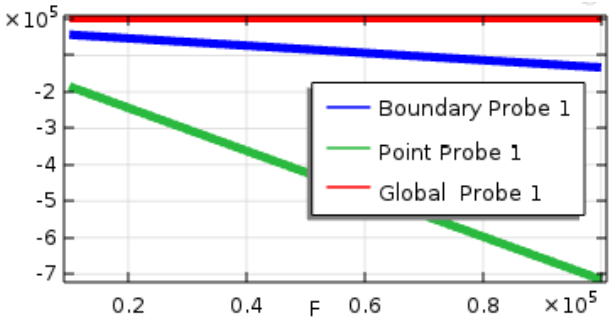
Probe Plot Group 14:

`Legend` – **Show legends, Position** = `True, Middle right`

Probe Plot Group 14 - Probe Table Graph 1:

`Legends` – **Show legends** = `True`

LMB > Plot



Зверніть увагу на те, що графік побудовано в окремому графічному вікні `Probe Plot 1`, а не в звичному вікні `Graphics`. Це видно з ярлика вікна графічної групи **Probe Plot Group 14**.

Підсумовуючи можна сказати, що перші три змінні, які створені з випадного списку доступних типів зондів, відслідковують інтегральні/зональні характеристики величин (інтеграл, середнє, екстремум) по області, поверхні або ребру. Зонди `Domain Point Probe` та `Boundary Point Probe` дозволяють вибрати точку всередині об'єму або на поверхні, і контролювати в ній значення виразу. `Global Variable Probe` зонд обчислює вираз, який не залежить від геометрії або точки.

Якщо ви редагували зонд і внесли до нього зміни, то клацніть по кнопці Update Results вгорі ПВ зонда, щоб оновити налаштування зонда і його дані. Потім клацніть правою кнопкою миші по вузлу **Definitions** і виконайте команду Update Probes, щоб перебудувати вторинні вузли.

Пам'ятайте, що всі зонди мають імена і відрізняються від звичайних змінних лише тим, що після обчислення моделі вони автоматично «створюють» ще кілька вузлів ДМ. Зонди можна використовувати в моделі як будь-які інші змінні, наприклад, в рівняннях, граничних умовах або в умовах зупинки обчислень.

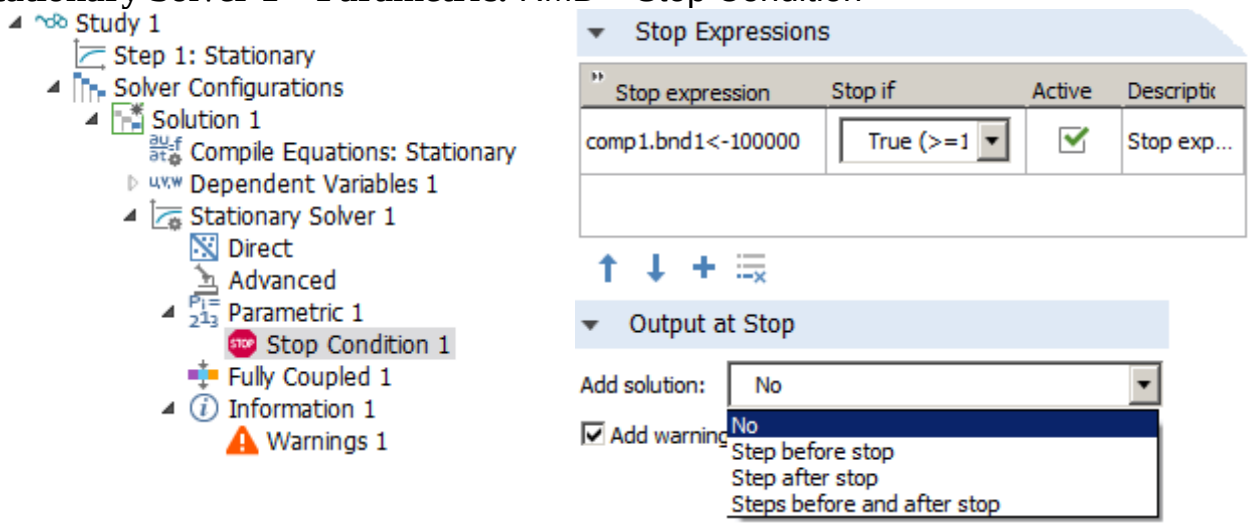
Зауваження. Зондові змінні можна створювати в динамічних задачах. В цьому випадку вони будуть залежати від часу, тобто будуть функціями параметра `t`.

Зупинка обчислень. Крім спостереження за змінними та зондами, ви можете використовувати їх для автоматичної зупинки обчислень. Зупиняти обчислення буває потрібно в різних випадках, наприклад, коли обчислення тривають занадто довго, або коли немає сенсу їх продовжувати (наприклад, виконалася умова, при якій речовина перейшла до фази руйнування). В COMSOL Multiphysics умови зупинки обчислень можна використовувати в трьох випадках: в динамічних задачах (time-dependent), при частотному аналізі (frequency-domain), та в статичних задачах при виконанні параметричного аналізу (parametric або auxiliary sweep).

Щоб передбачити автоматичну зупинку обчислень, по-перше треба мати скалярні змінні, що будуть використовуватися в логічному виразі, який буде постійно перевірятися. Наступним кроком є додавання підвузла **Stop Condition**. В разі параметричного аналізу він додається до вузла **Parametric** в гілці **Solver Configurations – Stationary Solver**. В разі динамічної задачі він додається до вузла **Time-Dependent Solver** в гілці **Solver Configurations**. Останнім кроком є введення в створеному вузлі **Stop Condition** умови зупинки обчислень. Вона є логічним виразом, який може набувати значення `true` або `false`.

Подивимось як можна зупинити обчислення, коли виконується допоміжний параметричний аналіз (Auxiliary Sweep). Нехай зупинка обчислень в нашій задачі відбувається, коли середній тиск на основу перевищує якийсь поріг, наприклад, більше `100000` Н/м^2. Нагадаємо, що середнє значення напруження σ_z по нижній грані тіла зберігається в зондовій змінній `bnd1`. З точністю до знака це є середній тиск на основу.

Розкрийте гілку **Study 1 - Solver Configurations – Solution 1 - Stationary Solver 1 – Parametric** і додайте в неї вузол `Stop Condition` командою **Stationary Solver 1 – Parametric:** RMB > Stop Condition



У вікні властивостей створеного вузла **Stop Condition** натисніть кнопку + Add, розташовану нижче розділу **`Stop Expressions`**, і в першому рядку таблиці в активованому полі стовпця **Stop expression** введіть вираз

`Stop Expressions` - **Stop expression** = `comp1.bnd1<-100000`

В другому стовпці відображається умова, при якій треба робити зупинку. Солвер буде обчислювати введений вираз після кожного кроку зміни параметра

і налаштування в стовпці **Stop if** визначають, що зупинка відбувається, коли логічний вираз істинний (`True`, ≥1), або хибний (`Negative`,<0).

Прапорець **Add warning** вказує, чи треба друкувати попередження, коли умова зупинки виконана.

Налаштування в списку **Add solution** визначають, що робити солверу при виконанні умови: залишати результати обчислення всіх «істинних» кроків, або всіх кроків, включно з тим, для якого умова порушилася. Нехай буде вибрана опція

Output at Stop - Add solution = `Step before stop`

Отже в нашій задачі обчислення повинні зупинятися, коли середнє напруження на нижній грані стане менше -100000 Н/м^2, тобто тиск на основу за модулем стане більше 100000 Н/м^2.

Виконайте команду

`Home->`Compute

Почнеться процес обчислень, який зупиниться при значенні параметра `F=60000`.

Щоб дізнатися значення, на якому було зупинено обчислення, треба відкрити ПВ вузла **Warnings 1** (див. попередній рисунок ліворуч). В розділі **Warnings** буде надруковано попередження з інформацією, на якому кроці солвер зупинився.

Warnings

Stop condition fulfilled for F=70000 (Stop expression 1).

Цю інформацію також можна знайти у вікні журналу/реєстру (Log). Звичайно, що скоротиться кількість рядків таблиці **Probe Table 2**.

F	Boundary Probe 1	Point Probe 1	Global Probe 1
10000	-42991	-1.8501E5	2300.0
20000	-52980	-2.4389E5	2300.0
30000	-62970	-3.0278E5	2300.0
40000	-72959	-3.6166E5	2300.0
50000	-82949	-4.2055E5	2300.0
60000	-92938	-4.7944E5	2300.0

Зверніть увагу на те, що відредаговані нами назви стовпців після обчислення моделі треба відновлювати.

Зауваження. Вираз, який використовується в умові зупинки (Stop expression) повинен розпізнаватися солвером за будь яких обставин. В ньому потрібно використовувати змінні з глобальним контекстом видимості. Тобто там можуть використовуватися імена глобальних змінних, або повні імена локальних змінних з префіксом – ім'ям компонента.

Залишимо в нашій моделі вузол **Stop Condition**, але деактивуємо його, щоб зупинка обчислень не відбувалася.

Study 1 - Solver Configurations – Solution 1 - Stationary Solver 1 – Parametric - Stop Condition 1: RMB > Disable

Підбір змінних. Коли ми кажемо про змінні задачі, то маємо на увазі не тільки ті, які створені у вузлах **Variables**. Часом нам потрібно мати змінні для визначення яких неможливо записати простий вираз. Їх треба знаходити, розв'язуючи алгебраїчні або диференціальні рівняння.

Поставимо таке питання, яким повинен бути зовнішній тиск на верхню грань, щоб середній тиск на основу дорівнював 100000 Н/м^2. Попередній параметричний аналіз з зупинкою обчислень показав, що він буде розташований в діапазоні від 60000 Н/м^2 до 70000 Н/м^2. Але як його знайти точніше.

Спочатку в нашій задачі відключимо параметричний аналіз.

Study 1 – Step 1: Stationary:

Study Extensions – **Auxiliary sweep** = False

і виконаємо обчислення командою

Home->Compute

Задача буде розв'язана при зовнішньому тиску 10000 Н/м^2, який визначається початковим значенням параметра F. Будуть також обчислені значення всіх зондів, але зараз вони вже не будуть функціями параметра, а будуть сталими величинами. Зондові криві в графічній групі **Probe Plot Group 14** перетворяться на точки.

Зауваження. Зонди можна створювати в статичних задачах без використання параметричного аналізу. В цьому випадку зонди являють сталі величини.

Після переобчислення моделі в таблиці **Probe Table 2** в першій колонці буде надруковано значення -42991 зондової змінної bnd1, яка містить середнє значення напруження σ_z на нижній грані тіла. Різниця між бажаним середнім значенням тиску на основу і реальним середнім значенням дорівнює

$$100000 - |\text{bnd1}| = 100000 - 42991 = 57009.$$

Нам потрібно зробити її нульовою.

Створимо змінну, яка буде містити шукане значення зовнішнього тиску. Назвемо її Fout. Цю, поки що невідому величину (тиск на верхню грань), треба використати в граничній умові у вузлі Boundary Load.

Відкрийте ПВ вузла **Component 1 - Solid Mechanics (solid) - Boundary Load 1** і введіть:

Force – $\mathbf{F_A}$ = 0,0,-Fout (*три поля*)

Оскільки зовнішній тиск Fout на верхню грань спрямований вниз (проти напряму вісі Z), то потрібно використати знак мінус.

Додамо в нашу фізичну модель рівняння відносно нової змінної Fout. Для цього виконайте команду

Physics **-> Global** – Global Equations

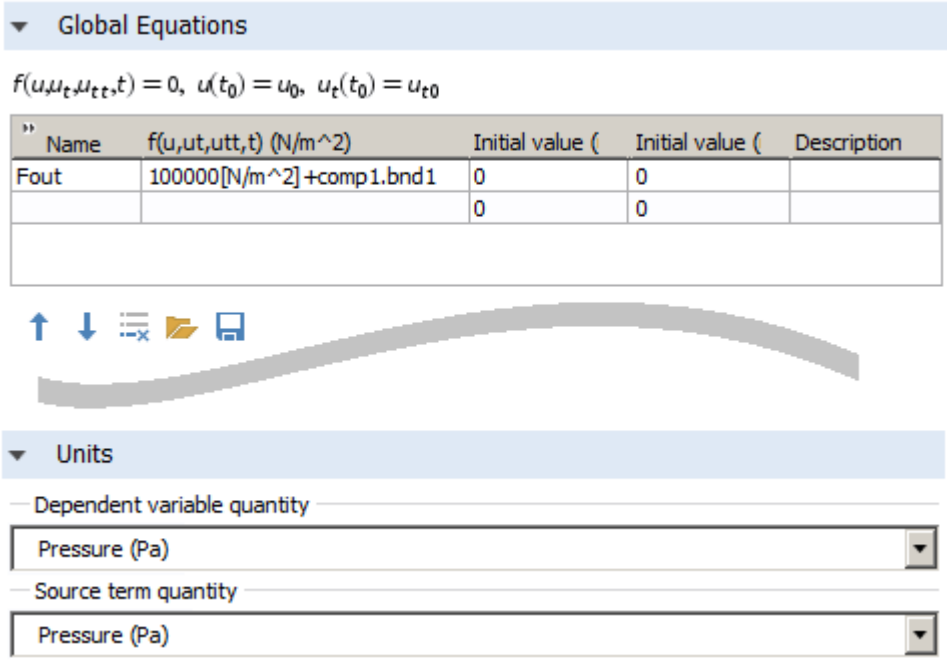
В ПВ створеного вузла **Solid Mechanics (solid) - Global Equations 1** в таблиці рівнянь введіть:

Global Equations – **Name** = Fout

Global Equations – **f(u,ut,utt,t)(1)** = 100000[N/m^2]+comp1.bnd1

Тут ми використали знак плюс тому, що змінна bnd1 насправді від'ємна.

Щоб введення стало доступним, потрібно клацнути по першому рядку в таблиці рівнянь. Потім в стовпці **Name** треба ввести ім'я змінної, а в другому стовпці – вираз, який повинен обернутися на нуль. Скоріш за все вираз буде виділено помаранчевим кольором. Це означає, що розмірності величин невизначені. Щоб задати розмірності змінної Fout і виразу f(u,ut,utt,t) заповніть розділ **Units** так, як показано на наступному рисунку.



Розмірність залежної (шуканої) змінної задається в полі **Units-Dependent variable quantity**, а розмірність виразу f(u,ut,utt,t) – в полі **Units-Source term quantity**.

В результаті ми додали в задачу рівняння $100000-|\mathrm{bnd1}(\mathrm{Fout})|=0$ і додаткову невідому Fout. Тепер виконайте команду

Home->Compute

Після обчислення в першому стовпці таблиці **Probe Table 2** буде стояти значення -1.0000E5 зондової змінної bnd1. Тобто середній тиск на основу дорівнюватиме 100000[N/m^2].

Щоб надрукувати значення шуканої невідомої Fout, виконайте інструкції:

Results -> **Derived Values** – Global Evaluation

Results - Derived Values - Global Evaluation 6:

Expression – **Expression** = comp1.Fout

LMB > Evaluate – New Table

Буде створена таблиця **Table 3** зі значенням змінної Fout=67070. Якщо це число використати в граничній умові для зовнішнього навантаження, то середній тиск на основу дорівнюватиме бажаному 100000 Н/м^2. По-перше, це зрозуміло зі значення зондової змінної bnd1, яке надруковано в таблиці **Probe Table 2**. По-друге, це можна перевірити, якщо зробити обчислення задачі при зовнішньому тиску на верхню грань 67070 Н/м^2.

Деактивуйте вузли **Derived Values - Global Evaluation 6** та **Solid Mechanics (solid) - Global Equations 1**. А в ПВ вузла **Solid Mechanics (solid) - Boundary Load 1** введіть

Force – $\mathbf{F_A}$ = 0,0,– 67070 (*три поля*)

Переобчисліть задачу командою

Study 1: RMB > Compute

Потім виконайте команду

Derived Values - Boundary Probe 1:

LMB > Evaluate – Table 3

В другому стовпці таблиці **Table 3** з'явиться бажане значення -1.0000E5 середнього напруження σ_z на нижній грані тіла. За модулем воно співпадає з середнім тиском на основу.

State variable Fout (Pa)	Stress tensor, z component (N/m^2), Boundary Prob
67070	-1.0000E5

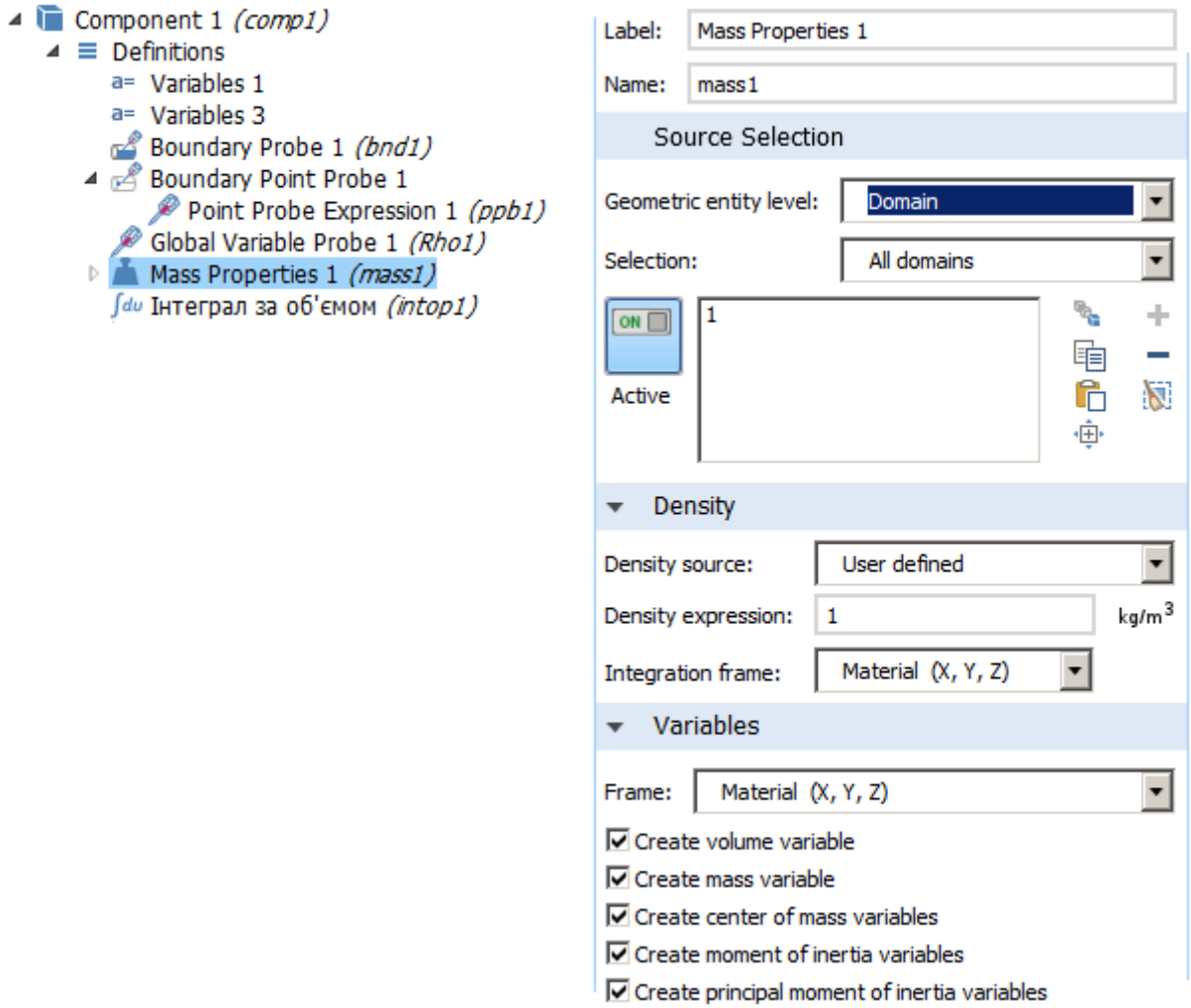
Тепер можете повернути модель в попередній стан, виконавши відкочення (скасування) останніх інструкцій деактивації, і повернувши в граничній умові вузла **Boundary Load 1** навантаження Fout.

Змінні маси (**mass properties**).

В цьому параграфі ми створили багато змінних, які зберігали такі розповсюджені властивості тіла, як об'єм та маса. Ці величини потрібні досить часто, і розробники системи це передбачили. Щоб створити такі змінні достатньо виконати команду

Definitions -> **Mass properties** – Mass properties

В гілці **Component – Definitions** створюється вузол **Mass Properties 1 (mass1)**, який буде представляти групу механічних і геометричних властивостей компонента. В ПВ створеного вузла в розділі **Source Selection** вказується рівень геометрії (Domain, Boundary, Edge або Point) і в секції вибору – множина, для якої будуть виконуватися обчислення. Для пошуку маси є розділ, в якому задається густина речовини. За наявності цього вузла можна обчислювати наступні характеристики: об'єм та масу тіла; центр мас; тензор інерції та його головні осі. Які характеристики обчислювати і зберігати в змінних моделі вказується відповідними прапорцями, що розташовані внизу ПВ вузла **Mass Properties 1(mass1)**.



Змінні маси стають доступними після виконання або оновлення обчислень.

Study 1: RMB > Update Solution

В результаті створюються наступні змінні:

Величина	**Ім'я змінної за замовчуванням**
Об'єм	`mass1.volume`
Маса	`mass1.mass`
Центр мас	`mass1.CMX, mass1.CMY, mass1.CMZ`
Момент інерції	`mass1.IXX, mass1.IXY, mass1.IXZ ...`
Головні осі інерції	`mass1.Ip1X, mass1.Ip2Y ...`
Головні моменти інерції	`mass1.Ip1, mass1.Ip2, mass1.Ip3`

Для тренування надрукуємо об'єм нашого тіла, обчисливши його за допомогою змінної `mass1.volume`.

`Results` -> **`Derived Values`** – Global Evaluation

Results - Derived Values - Global Evaluation 7:

`Expression` – **Expression** = `comp1.mass1.volume`

LMB> Evaluate – Table 1

comp1.intop1(1) (m^3)	comp1.Vol1 (m^3)	comp1.Mass1 (kg)	Force (N)	Volume (m^3)
4.7019	4.7019	10814	1.3820E5	4.7019

В перших комірках таблиці **Table 1** вже було надруковано значення об'єму і воно співпадає з тим, що ми зараз отримали.

Зауваження. В компоненті може бути кілька вузлів `Mass Properties`.

Підведемо короткий підсумок. В локальних вузлах **Definitions** створюються конструкції, які допомагають працювати з компонентом: змінні, функції, оператори, зонди, маси. Всі ці об'єкти мають імена, і звернення до них виконується з використанням точкової нотації. Щоб прийняти додані визначення до уваги потрібно оновити обчислення.

Скалярні змінні можна додавати в модель як невідомі алгебраїчних або звичайних диференціальних рівнянь, які формуються у вузлах `Global Equations`.

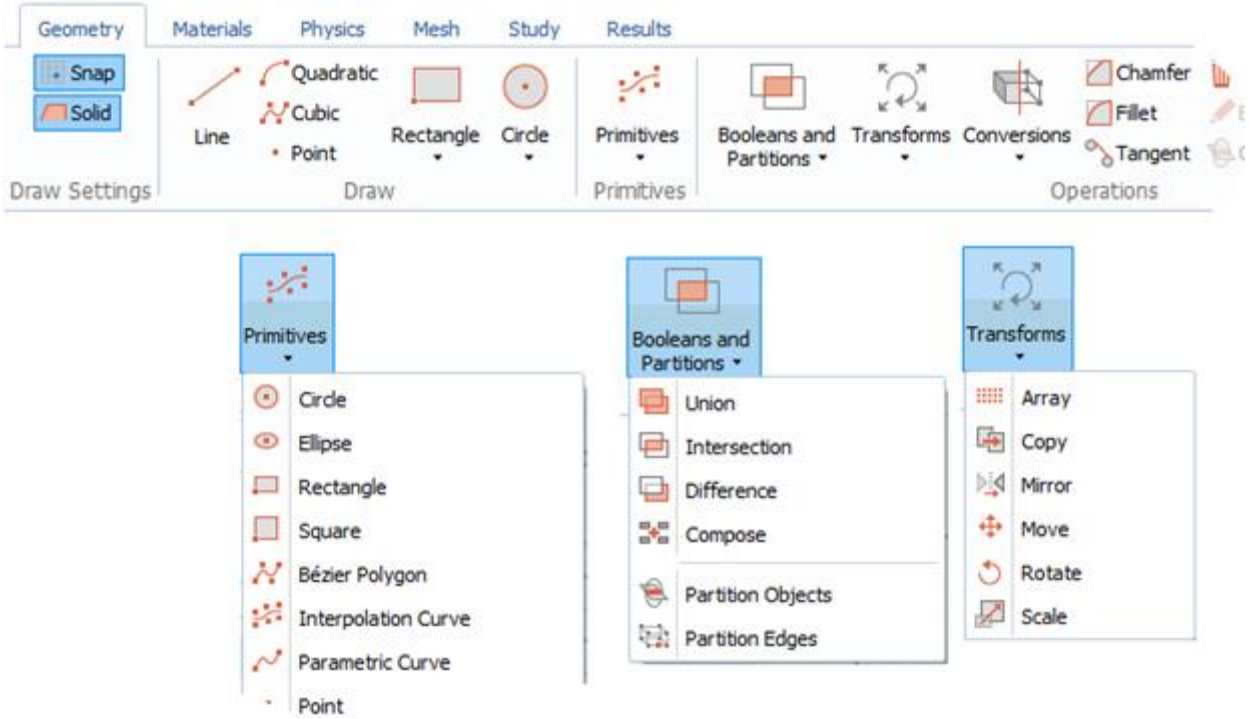
1.5.5. Побудова геометрії.

Побудова розрахункової області, яку ми часто називаємо геометрією, є одним з перших кроків моделювання в COMSOL Multiphysics. В системі існує багато інструментів та операцій, призначених для цього.

Головними кроками створення геометрії є:

- креслення геометричних фігур (їх набір залежить від просторової розмірності);
- використання різноманітних операцій (булевих, перетворення та ін.) для створення нових фігур з вже побудованих;
- визначення способу використання сполучених геометричних об’єктів за допомогою завершальної операції Form Union або Form Assembly.

Всі інструменти, які призначені для створення геометрії, зосереджені на стрічці `Geometry`. Для двовимірних моделей основна частина стрічки `Geometry` показана на наступному рисунку. Там же зображено вміст трьох найбільш часто використовуваних груп команд (меню): **`Primitives`**, **`Booleans and Partitions`** та **`Transforms`**. З можливостями деяких з них ви вже познайомилися.



Основою конструювання геометрії є «солідні» (solid) об'єкти. Цей термін буде використовуватися стосовно фігур на площині, що мають скінченну площу, і до тіл в просторі, які мають скінченний об’єм. Солідні об'єкти складаються з граничної і внутрішньої частини і повністю визначаються в термінах власних границь. Іноді солідний об'єкт має внутрішні межі, що розділяють його підобласті, в яких можуть бути задані різні матеріальні властивості або СкЕ сітка.

Найпростіший спосіб конструювання геометрії складається в виборі об’єкта з вбудованого списку геометричних фігур, які зазвичай називають примітивами. При виборі команди з групи примітивів (**Primitives**) система створює вузол ДМ і додає креслення фігури в геометрію. Після цього в ПВ створеного вузла можна налаштувати розміри і положення фігури.

Інший спосіб створення геометричних об’єктів, доступний в 2D і 1D моделях, полягає в кресленні примітивів за допомогою миші. Відповідні інструкції розташовані в групі команд **Draw**. Вони виконуються наступним чином:

- клацніть по кнопці фігури, яку слід створити;
- натисніть ліву кнопку миші в графічному вікні для визначення центра або кута фігури;
- протягніть мишу (з натиснутою кнопкою), задавши бажаний розмір, і відпустіть мишу; фігура негайно з’явиться в графічному вікні, а її вузол в дереві моделі;
- якщо треба, відредагуйте властивості фігури у ПВ створеного вузла.

Після побудови кількох фігур можна почати комбінувати їх багатьма способами, використовуючи геометричні операції. Булеві (boolean) операції об’єднання (union), перетину (intersection) та різниці (difference), використовуються для комбінування фігур, створюючи з них нові більш складні об’єкти. Операції перетворення (transformation) - Move, Copy, Mirror та Array дозволяють змінити положення (Move) фігури, створити її копію (Copy), дзеркально відобразити відносно точки, лінії чи площини (Mirror), або розмножити її копії у вигляді масиву (Array).

В 3D моделях ви не можете креслити геометричні фігури мишею, але можете створювати поперечні перерізи на робочих площинах, які потім можете поширювати в 3D об’єкти за допомогою операцій перетворення:

- операція видавлювання (Extrude) дозволяє поширити об’єкт, що створений в робочій площині або на плоскій грані, в тривимірне тіло;
- операція обертання (Revolve) навколо вісі створює з 2D фігури тривимірне тіло;
- операція розгортання (Sweep) виконує перенесення з замітанням 2D фігури вздовж криволінійного шляху, створюючи тривимірний об’єкт.

Інший підхід полягає в побудові межі тривимірного тіла. В цьому випадку ви явно конструюєте замкнуту поверхню твердого тіла, прорисовуючи кожну його грань. Потім така поверхня перетворюється в тверде тіло. Цей спосіб називається граничним (або поверхневим) моделюванням. Граничне

моделювання неявно використовується при двовимірному кресленні, коли включений Solid режим. Якщо на інструментальні панелі Geometry (в 2D режимі) натиснута кнопка Solid Solid, то при кресленні на площині замкнутих кривих утворюються двовимірні солідні об’єкти.

Звісно, що розміри та положення фігур можна задавати, використовуючи глобальні параметри.

Всі дії по створенню геометрії відображаються в гілці **Geometry** дерева моделей, яка фактично містить список інструкцій креслення розрахункової області компонента. При конструюванні геометрії система виконує побудови з цього списку в послідовності зверху вниз. Отже крім створення елементів геометрії в гілці **Geometry** визначається послідовність їх побудови і комбінування. Безпосередньо ПВ самого вузла **Geometry** містіть налаштування одиниць вимірювання довжин та кутів.

При роботі з геометричною гілкою дерева моделі можна використовувати стандартні операції роботи з буфером: копіювання і вставку вузлів-інструкцій. Але скопійований об'єкт може бути вставлений тільки в геометричну послідовність ідентичної розмірності.

Останній вузол **Form Union/Assembly** гілки **Geometry** визначає завершальний спосіб формування геометрії.

- За замовчуванням використовується створення об'єднання (union). Тоді система виконує логічне об'єднання всіх об'єктів послідовності, тобто створює суцільне тіло. Інколи воно може бути поділене на області, відокремлені внутрішніми межами, які призначені, наприклад, для задання різних матеріальних властивостей об’єднаних частин.
- Альтернативою є збірка (assembly). Тоді система інтерпретує геометрію як колекцію об'єктів (замість їх об'єднання). Це означає, що користувач сам буде визначити способи зв’язку розрізнених тіл за допомогою інструментів, які називаються «парами» (pairs). Пари, наприклад, будуть визначати межі, де шукане поле неперервно. За замовчуванням пари створюються автоматично при формуванні збірки.

В гілці **Geometry** після вузла **Form Union/Assembly** можуть бути розташовані лише вузли віртуальних операцій, які використовують існуюче геометричне представлення як базове для конструювання альтернативної геометрії виключно до цілей дискретизації і постановки фізичної задачі.

Кнопка Build All в вгорі ПВ геометричного вузла будує всі об'єкти поточної геометрії.

Приклад 1. Побудуємо тіло, яке зображено на наступному рисунку.

Створіть порожню модель.

File->New->Blank Model

Додайте в модель тривимірний компонент.

Untitled.mph (root) : RMB > Add Component ->3D

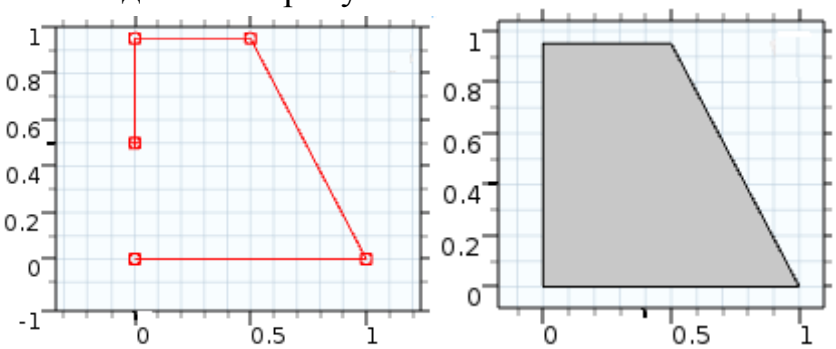
З'явиться гілка **Component 1** з вузлом **Geometry 1**. Креслення фігури в цьому вузлі складається з кількох кроків.

1. *Створення робочої площини*. При активному вузлі **Geometry 1** виконайте команду

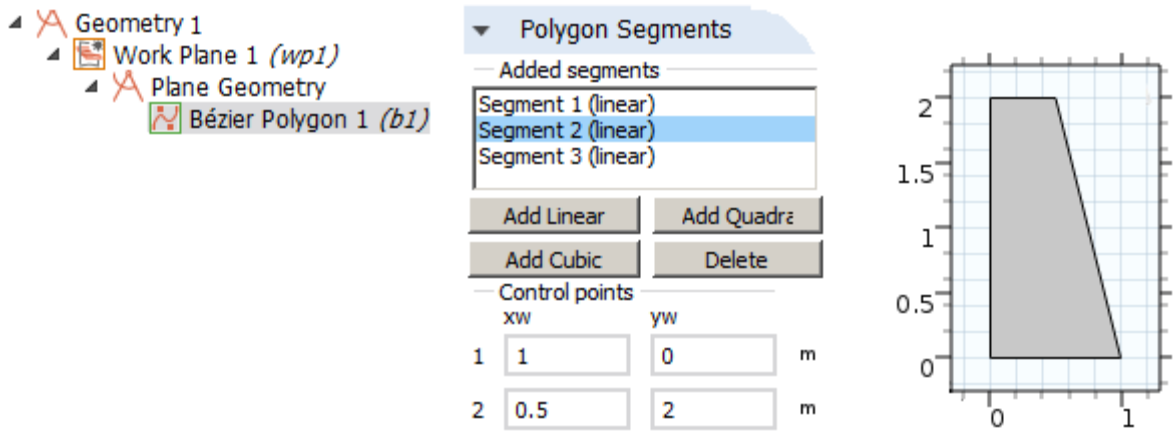
Geometry->**Work Plane**->Work Plane

В ПВ створеного вузла **Work Plane 1** всі параметри залиште без змін. Вони відповідають тому, що робоча площина буде співпадати з площиною XY. Клацніть по кнопці Show Work Plane вгорі панелі. Цим ви активуєте вузол **Work Plane 1 – Plane Geometry**.

2. *Креслення фігури в робочій площині*. На закладці Work Plane інструментальної панелі перевірте, що натиснуті кнопки Snap і Solid. Кнопка Snap включає режим «притягання» миші до точок графічної сітки (grid). На панелі в секції **Draw** натисніть кнопку Line, переведіть покажчик миші в графічну область і намалюйте багатокутник, показаний на наступному рисунку праворуч. Закінчіть креслення ламаної (коли вона буде вже замкненою) клацанням правою кнопкою миші в зоні малюнка. Крива автоматично перетвориться на солідний чотирикутник.



Відредагуйте координати його вершин так, щоб вони дорівнювали (0,0), (1,0), (0.5,2), (0,2). Для цього в розділі **Polygon Segments** послідовно виділяйте сегменти ламаної і редагуйте значення координат вершин в полях **Control points**.



Зверніть увагу на те, що вузол ДМ отримав назву **Bézier Polygon 1 (b1).**

Додайте до багатокутника фаску і скруглення. Для цього виконайте команди

`Work Plane->`**`Operations`**`->` Chamfer

Work Plane 1 (wp1) - Plane Geometry - Chamfer 1 (cha1):

`Points` – **Vertices to chamfer** = `b1:4` (*права нижня вершина*)

`Distance` – **Distance from vertex** = `0.25` (*розмір фаски*)

RMB > Build Selected (*наступний рисунок ліворуч*)

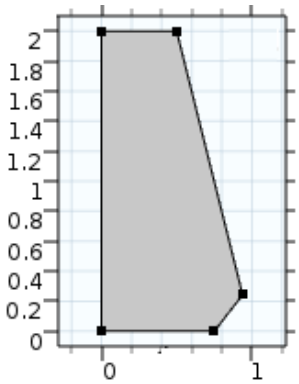
`Work Plane->`**`Operations`**`->` Fillet

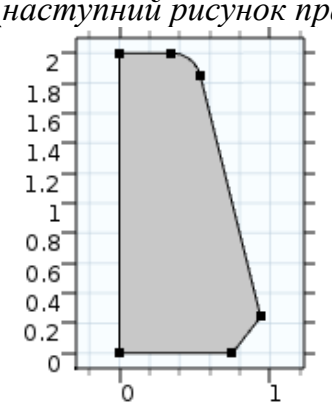
Work Plane 1 (wp1) - Plane Geometry - Fillet 1 (fil1):

`Points` – **Vertices to fillet** = `cha1:3` (*права верхня вершина*)

`Radius` – **Radius** = `0.2` (*радіус скруглення*)

RMB > Build Selected (*наступний рисунок праворуч*)





Створіть профіль ребер. Для цього побудуйте прямокутник, а з нього – масив прямокутників. Потім цей масив видаліть з вже побудованої фігури.

`Work Plane->`**`Primitives`**`->` Rectangle

Work Plane 1 (wp1) - Plane Geometry - Rectangle 1 (r1):

`Size` – **Width, Height** = `0.5,0.25` (*два поля; розміри*)

`Position` – **xw, yw** = `0.5,0.4` (*два поля; положення кута*)

RMB > Build Selected (*наступний рисунок ліворуч*)

`Work Plane->`**`Transforms`**`->` Array

Work Plane 1 (wp1) - Plane Geometry - Array 1 (arr1):

`Input` – **Input objects** = `r1` (*фігура для розмноження*)

`Size` – **xw size, yw size** = `1,3` (*кількість стовпців і рядків масиву*)

`Displacement` – **xw, yw** = `0,0.5` (*зміщення елементів масиву*)

RMB > Build Selected (*наступний рисунок всередині*)

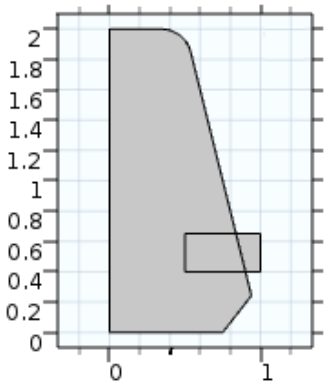
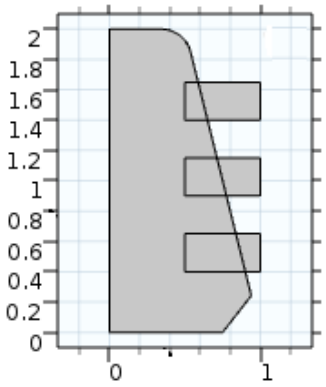
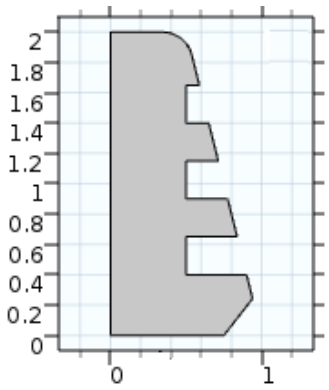
`Work Plane->`**`Booleans and Partitions`**`->` Difference

Work Plane 1 (wp1) - Plane Geometry - Difference 1 (dif1):

`Difference` – **Objects to add** = `fil1`

`Difference` – **Objects to subtract** =`arr1(1,1),arr1(1,2),arr1(1,3)`

RMB > Build Selected (*наступний рисунок праворуч*)

Створіть профіль отвору.

`Work Plane->`**`Primitives`**`->` Rectangle

Work Plane 1 (wp1) - Plane Geometry - Rectangle 2 (r2):

`Size` – **Width, Height** = `0.6,1.5` (*два поля; розміри*)

`Position` – **xw, yw** = `-0.3,0` (*два поля; положення кута*)

`Work Plane->`**`Primitives`**`->` Circle

Work Plane 1 (wp1) - Plane Geometry – Circle 1 **(c1):**

`Size and Shape` – **Radius** = `0.3`

`Position` – **Base, xw, yw** = `Center;0,1.5` (*три поля*)

RMB > Build Selected (*наступний рисунок ліворуч*)

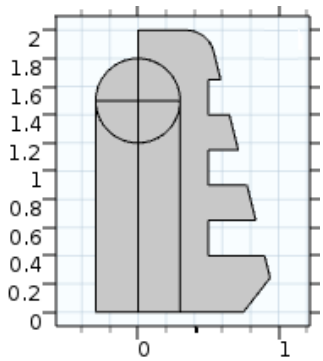
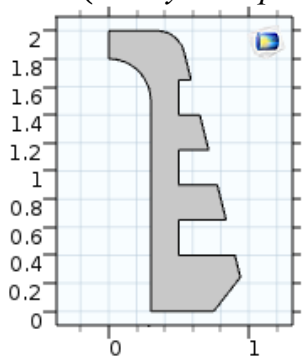
`Work Plane->`**`Booleans and Partitions`**`->` Difference

Work Plane 1 (wp1) - Plane Geometry - Difference 2 (dif2):

`Difference` – **Objects to add** = `dif11`

`Difference` – **Objects to subtract** =`r2,c1`

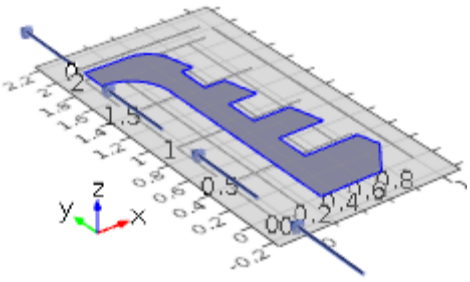
RMB > Build Selected (*наступний рисунок праворуч*)

3. *Створення тіла обертання*. Операція Revolve призначена для утворення тіла обертання з профілю, який побудований в робочій площині. Для неї додатково треба вказати орієнтацію вісі обертання.

Work Plane 1 (wp1):

`Geometry ->`**`Operations`**`->` Revolve



Geometry 1 – Revolve 1 (rev1):

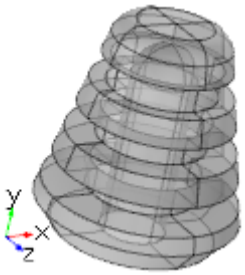
General – **Input objects** = wp1 (*робоча площина*)

Задання в робочій площині вісі обертання за допомогою точки та вектора (двовимірного) напрямку.

Revolution Axis – **Points on the revolution axis – xw, yw** = 0,0

Revolution Axis – **Direction of revolution axis – xw, yw** = 0,1

RMB > Build Selected (*наступний рисунок повернуто*)



Зауваження. При побудові тіла обертання вісь обертання не повинна пересікати контур фігури.

4. *Надбудова тривимірних елементів*. Додамо зверху до тіла прямокутну призму. Для цього верхню грань побудованого тіла використаємо в якості нової робочої площини. На ній накреслимо квадрат, який «витягнемо» вгору для утворення призми.

Geometry->**Work Plane**->Work Plane

Work Plane 2 (wp2):

Plane Definition – **Plane type** = Quick (*спосіб вибору площини*)

Plane Definition – **Plane** = zx-plane

Plane Definition – **Offset type, y-coordinate** = Distance; 2

(*наступний рисунок ліворуч*)

Work Plane 2 (wp2) - Plane Geometry:

Work Plane->**Primitives**-> Square

Work Plane 2 (wp2) - Plane Geometry - Square 1 (sq1):

Size – **Side length** = 0.3

Position – **Base, xw, yw** = Center;0,0 (*три поля*)

RMB > Build Selected (*наступний рисунок всередині*)

Операція Extrude призначена для видавлювання профілю робочої площини в простір для побудови тривимірного тіла. Зміщення профілю за замовчуванням виконується в напрямку нормальному до площини. Додатково слід вказати відстань.

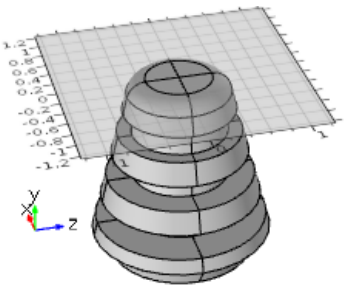
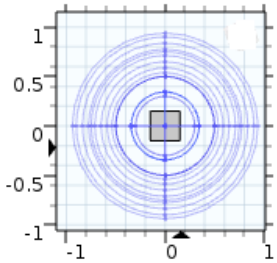
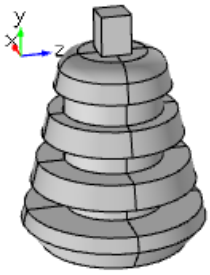
Work Plane 2 (wp2): Geometry ->**Operations**-> Extrude

Geometry 1 – Extrude 1 (ext1):

General – **Input objects** = wp2 (*робоча площина*)

Distance from Plane – **Distance** = 0.4 (*відстань видавлювання*)

RMB > Build Selected (*наступний рисунок праворуч*)

Додамо зверху кільце, яке проходить крізь призму. Кільце побудуємо за допомогою операції Sweep (розгортка), яка переносить 2D фігуру вздовж шляху (кривої) в просторі, замітаючи тривимірну область і утворюючи 3D тіло. Для цього нам потрібно створити 2D фігуру (коло) і криву, вздовж якої буде виконуватися перенесення. Коло створимо в робочій площині XY, а криву – в площині YZ. Звісно, що ці робочі площини теж треба створити.

Створіть коло.

`Geometry->`**`Work Plane`**`->Work Plane`

Work Plane 3 (wp3):

`Plane Definition` – **Plane** = `xy-plane`

`Plane Definition` – **Offset type, y-coordinate** = `Distance; 0`

Work Plane 3 (wp3) - Plane Geometry:

`Work Plane->`**`Primitives`**`-> Circle`

Work Plane 1 (wp1) - Plane Geometry - Circle 1 (c1):

`Size and Shape` – **Radius** = `0.1`

`Position` – **Base, xw, yw** = `Center; 0, 2.2` (*три поля*)

RMB > Build Selected (*наступний рисунок ліворуч*)

Створіть криву перенесення.

Geometry 1: `Geometry->`**`Work Plane`**`->Work Plane`

Work Plane 4 (wp4):

`Plane Definition` – **Plane** = `yz-plane`

`Plane Definition` – **Offset type, y-coordinate** = `Distance; 0`

Work Plane 4 (wp4) - Plane Geometry:

`Work Plane->`**`Primitives`**`-> Circle`

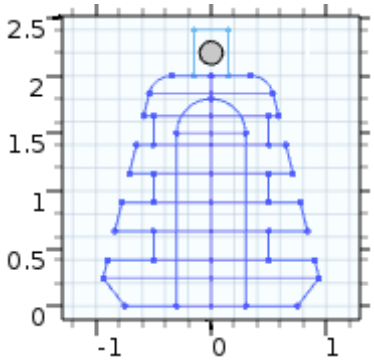
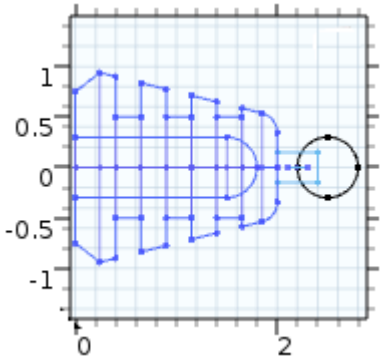
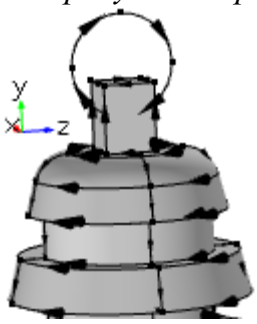
Work Plane 4 (wp4) - Plane Geometry - Circle 1 (c1):

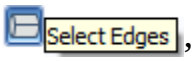
`Object Type` – **Type** = `Curve` (*створюємо окружність, а не коло*)

`Size and Shape` – **Radius** = `0.3`

`Position` – **Base, xw, yw** = `Center; 2.5, 0` (*три поля*)

RMB > Build Selected (*наступний рисунок всередині*)

Розгортку неможна виконати по замкненій кривій. Тому для побудови кільця створимо дві розгортки по половинах окружності в різні боки. Слід мати на увазі, що перенесення виконується вздовж напрямку зростання параметра кривої. Його можна наочно побачити, якщо у вузлі **Component 1 (comp1) – View 1** включити опцію **Show edge direction arrows**, а в графічному вікні кнопкою 

, розташованою над цим вікном, включити режим Select Edges.

Work Plane 4 (wp4): RMB > Build Selected (*попередній рисунок праворуч*)

Створіть розгортки.

`Geometry` ->**`Operations`**-> Sweep

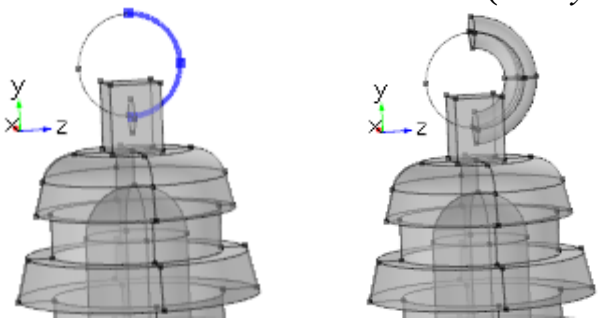
Geometry 1 – Sweep 1 (swe1):

`Cross Section` – **Faces to sweep** = `wp3:1`(*площина з круговим перерізом*)

`Spline Curve` – **Edges to follow**= `wp4:3,4` (*наступний рисунок ліворуч*)

`Spline Curve` – **Reverse direction**= `True` (*в зворотному напрямку*)

RMB > Build Selected (*наступний рисунок праворуч*)



Зверніть увагу на опцію **Reverse direction** = `True`, яка включає перенесення кола в оберненому напрямку вздовж кривої (інакше перенесення виконується не в той бік).

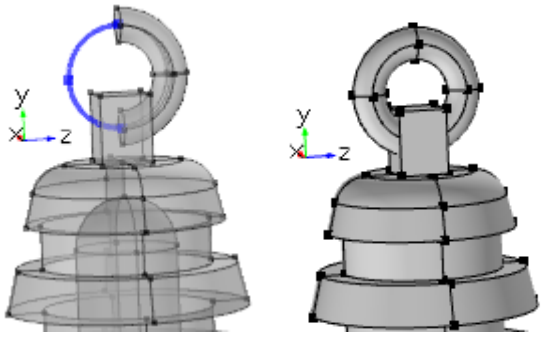
`Geometry` ->**`Operations`**-> Sweep

Geometry 1 – Sweep 2 (swe2):

`Cross Section` – **Faces to sweep** = `wp3:1`

`Spline Curve` – **Edges to follow**= `wp4:1,2` (*наступний рисунок ліворуч*)

RMB > Build Selected (*наступний рисунок праворуч*)



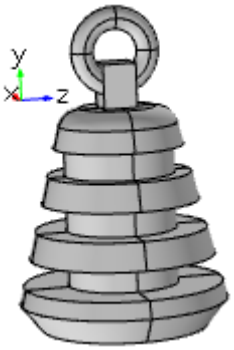
5. *Визначення завершального способу формування геометрії.* Останнім вузлом в гілці **Geometry 1** залиште **Form Union (fin)**.

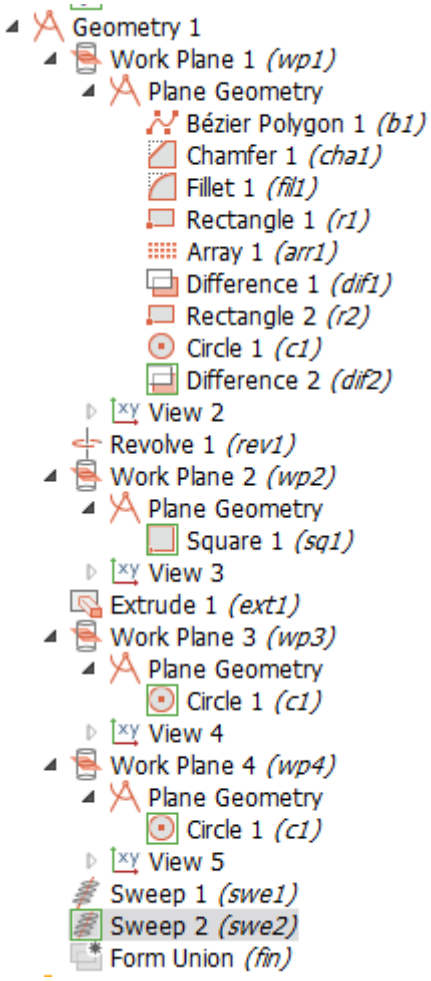
RMB > Build All Objects (*наступний рисунок ліворуч*)

Коли ви побажаєте розглядати кільце як рухому частину конструкції, наприклад, в задачі динаміки тіл (Multibody Dynamics), то в ПВ цього вузла в

списку **Action** слід обрати значення `Form an assembly`. Але тоді в верхній призмі потрібно буде ще спроектувати отвір під кільце.

Якщо ви виконували інструкції в послідовності з нашим прикладом, то гілка **Geometry 1** ДМ буде мати вигляд, зображений на наступному рисунку праворуч.

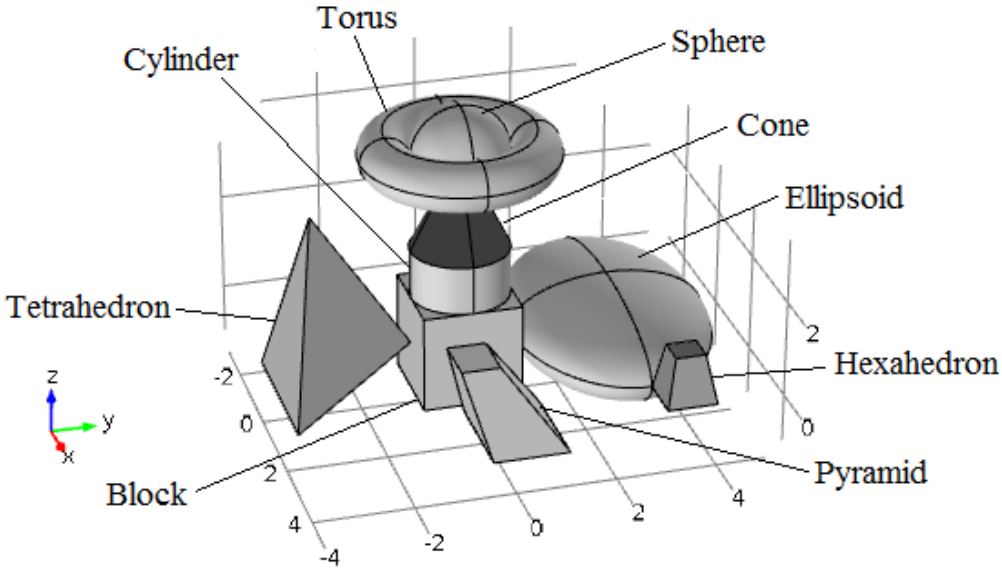




Зауваження. Кільце можна було б створити з використанням примітива `Torus` (тор). ■

Ми не описували процес створення тривимірних графічних примітивів. Потренуйтесь в цьому самостійно.

Вправа. Використовуючи тривимірні графічні примітиви, створіть збірку, яка зображена на наступному рисунку (звісно без виносок).



Команди/кнопки побудови тривимірних примітивів розташовані на закладці `Geometry` інструментальної панелі.

Оскільки не всі тіла на рисунку перетинаються, то створити з них єдине тіло неможливо. Набір неперетинних солідних об'єктів називається збіркою (assembly). Щоб її створити в гілці **Geometry** в її останньому вузлі **Form Union** в полі **`Form Union/Assembly`** - **Action** слід вибрати значення `Form an assembly`. Інакше при виконанні команди Build All ви отримаєте повідомлення про помилку.

1.6. Демонстраційні приклади.

1.6.1. Охолодження куба

Розв'яжемо задачу поширення тепла в одиничному кубі, поверхня якого підтримується при нульовій температурі, а початкова температура дорівнює 1.

Створіть модель послідовністю команд:

File->New->Model Wizard->3D->

Heat Transfer - Heat Transfer in Solids (ht) ->Time Dependent

Будемо розв'язувати безрозмірні рівняння. Для цього в ПВ вузла **Component 1 (comp1)** введіть:

`General` - **Unit system** = `None`

Геометрія моделі складається з одного куба. Намалюємо його.

При активному вузлі **Geometry 1** виконайте послідовність команд:

`Geometry`**`->Pimitives->`**Block

За замовчуванням параметри блоку збігаються з тими, які нам потрібні. Не міняємо їх. Завершіть креслення куба клацанням по кнопці Build All Objects.

Відкрийте ПВ вузла **Heat Transfer in Solids(ht) - Heat Transfer in Solids 1**. В розділі **`Equation`** цієї панелі наведено загальний вигляд нестаціонарного рівняння теплопровідності

$$\rho C_\rho \frac{\partial T}{\partial t} + \rho C_\rho \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = Q + Q_{ted},$$

де T - шукана функція температури, $\mathbf{q} = -k\nabla T$ – тепловий потік, k - коефіцієнт теплопровідності, $\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]$ – оператор градієнта. Поле швидкостей **u** в нашій задачі не використовується, так само як і джерела тепла, які представлені в правій частині рівняння. За замовчуванням вони всі дорівнюють нулю. Тому рівняння має вигляд $\rho C_\rho \frac{\partial T}{\partial t} - k(\nabla \cdot \nabla T) = 0$.

В ПВ вузла **Heat Transfer in Solids 1** введіть:

`Heat Conduction,Solid` – **Thermal conductivity -k**= `User defined`
– **Thermal conductivity -<>**= `1`

`Thermodynamics,Solid` – **Density – ρ, <>** = `User defined,1`

`Heat capacity at constant pressure` – C_ρ, **<>**= `User defined,1`

Перевірте, що в ПВ вузла **Heat Transfer in Solids 1** в розділі **`Domain Selection`** в секції **Selection** обрана область з номером 1.

Задайте початкову температуру. Для цього в ПВ вузла **Initial Values 1** введіть:

`Initial Values` – **`Temperature`** – **T** = `1`

Задайте граничну температуру. Для цього виконайте послідовність команд:

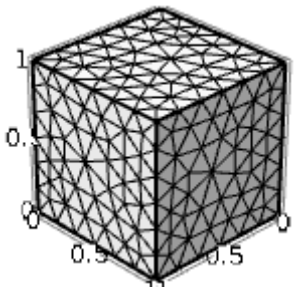
`Physics->`**`Boundaries`** ->Temperature

Активуйте вузол **Temperature 1**, натисніть кнопку Active в розділі **`Boundary Selection`** і в списку **Selection** виберіть `All boundaries`. Номери всіх граней `1,2,3,4,5` і `6` додадуться до списку **Selection**. В розділі **Temperature** в поле $\mathbf{T_o}$ введіть `0`.

Побудуйте СкЕ сітку. Для цього в ПВ вузла **Mesh 1** введіть:

`Mesh Settings` –**Element size** = `Normal`

і натисніть кнопку Build All. У графічному вікні ви побачите зображення СкЕ сітки в області куба.



Зауважимо, що тривимірні обчислення тривають значно довше двовимірних. Тому не намагайтеся відразу будувати дрібну сітку. Виконайте всі обчислення для «нормальної» сітки і потім, якщо буде потреба, подрібніть сітку.

В дереві моделі активуйте вузол **Step 1: Time Dependent**. В його ПВ задайте послідовність моментів часу, в яких буде обчислюватися розв'язок:

`Study Settings` –**Times** = `0.001 0.005 range(0.01,0.01,0.1)`

Розв'яжіть задачу, виконавши команду

`Study`**->** Compute.

В дереві моделі в гілці **Results** з'явиться гілка **Temperature (ht)** з вузлом **Surface 1**, що містить функціональне забарвлення поверхні куба, яке представляє температуру. З'явиться також гілка **Isothermal Contours (ht)**, яка містить контурний графік температури на поверхні куба. Обидва графіка неінформативні, оскільки на поверхні куба у нас температура незмінна і дорівнює 0. Видалимо ці графіки. Клацніть правою кнопкою миші по вузлу **Temperature (ht)** і в випадному меню виберіть команду Delete. Аналогічно видаліть гілку **Isothermal Contours (ht)**.

Побудуємо графік розподілу температури всередині куба. Створіть тривимірну графічну групу `3D Plot Group` командою

`Results->`3D Plot Group

В ПВ створеного вузла **3D Plot Group 3** введіть:

`Data`– **Data set** = `Study 1/Solution 1`

`Data`– **Time (s)** = `0.02`

`Color Legend` – **Show legends** =`Включити`

В графічній групі **3D Plot Group 3** створіть підвузол `Multislice`. В ПВ створеного вузла **3D Plot Group 3 - Multislice 1** введіть:

`Data`– **Data set** = `From parent`

`Expression`– **Expression** = `T`

В розділі **`Multiplane Data`** в полях **x-planes**, **y-planes**, **z-planes** вкажіть кількість площин – по одній площині кожної орієнтації.

`Range` – **Manual color range** = `Включити`

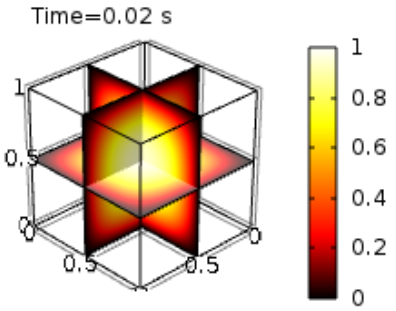
– **Minimum, Maximum** = `0 , 1`

`Coloring and Style` – **Coloring** =`Color table`

– **Color** =`Thermal`

– **Color legend** = `Включити`

Виконайте команду Plot і ви отримаєте зображення трьох взаємно ортогональних площин, що проходять через центр куба, з функціональним забарвленням, яке відповідає температурі куба в момент часу `t = 0.02`.



Змінюючи в ПВ вузла **3D Plot Group 3** значення в полі **Data - Time (s)**, ви будете отримувати аналогічні графіки для інших моментів часу.

В динамічних задачах можна будувати графіки залежності розв’язку від часу в точках. Для цього треба використовувати графічні вузли типу **Point Graph**, які слід прив’язувати до точкового джерела даних `Cut Point 3D` або просто до точок, що вже існують в геометрії. Якщо в ПВ такого вузла в списку

Time selection вибрати `All`, то буде побудовано часовий графік по значенням розв'язку в точці в усі моменти часу.

Побудуємо графік залежності температури від часу в центрі куба. Для цього спочатку створимо точкове джерело даних, виконавши інструкції

`Results->`**`Data Set`**`->`Cut Point 3D

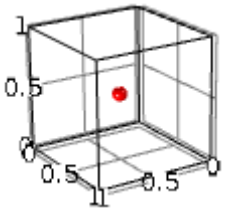
В ПВ створеного вузла **Cut Point 3D 1** введіть:

`Data` – **Data set** = `Study 1/Solution 1`

`Point Data` –**Entry method** = `Coordanates`

`Point Data` – **x, y, z** = `0.5, 0.5, 0.5` (*три поля*)

Перевірте введення точки командою Plot.



Створіть групу одновимірних графіків командою

`Results ->` 1D Plot Group

В ПВ створеного вузла **1D Plot Group 4** введіть:

`Data` – **Data set** = `Cut Point 3D 1`

`Data` – **Time selection** = `All`

На стрічці панелей з'являється закладка `1D Plot Group 4`. З неї виберіть команду

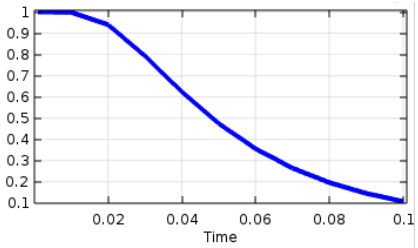
`1D Plot Group 4->` Point Graph

В ПВ створеного вузла **Point Graph 1** введіть:

`y-Axis Data` – **Expression** = `T`

`x-Axis Data` – **Parameter** = `Time`

Побудуйте графік залежності температури в точці від часу командою Plot. Якщо треба, то налаштуйте колір і стиль кривої в розділі **`Coloring and Style`** ПВ вузла **Point Graph 1**.



Надрукуємо значення температури в центрі куба в різні моменти часу. Для цього у вузлі **Derived Values** створіть вузол типу `Point Evaluation` наступною командою

Derived Values: RMB > Point Evaluation

В ПВ створеного вузла **Point Evaluation 1** введіть:

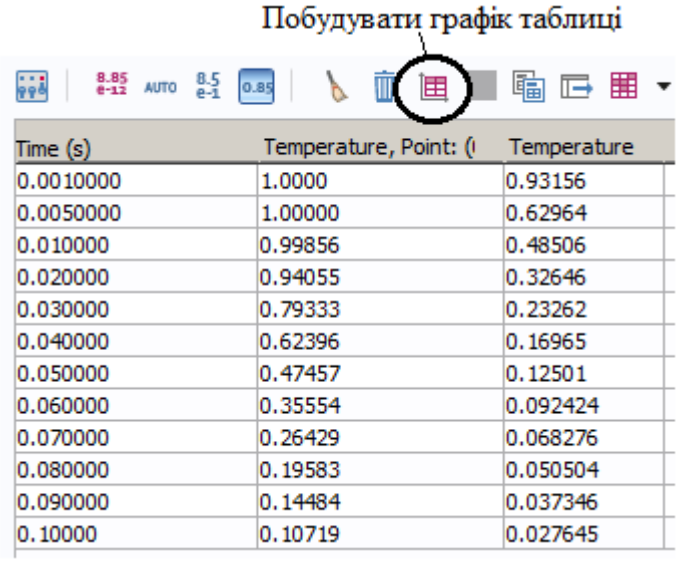
`Data` – **Data set** = `Cut Point 3D 1`

`Data` – **Time selection** = `All`

`Expression` – **Expression** = `T`

LMB > Evaluate – New Table

В інформаційній зоні буде створена таблиця **Table 1** така, як на наступному рисунку, але без правого стовпця.



Time (s)	Temperature, Point: (	Temperature
0.0010000	1.0000	0.93156
0.0050000	1.00000	0.62964
0.010000	0.99856	0.48506
0.020000	0.94055	0.32646
0.030000	0.79333	0.23262
0.040000	0.62396	0.16965
0.050000	0.47457	0.12501
0.060000	0.35554	0.092424
0.070000	0.26429	0.068276
0.080000	0.19583	0.050504
0.090000	0.14484	0.037346
0.10000	0.10719	0.027645

Вона містить значення температури в обраній точці в моменти часу, які задані у вузлі **Step 1: Time Dependent** в полі **`Study Settings`** –**Times**. Ці моменти `0.001 0.005 0.01 ... 0.09 0.1` відображені в лівому стовпці таблиці.

Побудуємо графік середньої температури куба в залежності від часу. Для цього виконайте інструкції

Derived Values: RMB > Average - Volume Average

В ПВ створеного вузла **Volume Average 1** введіть:

`Data` – **Data set** = `Study 1/Solution 1`

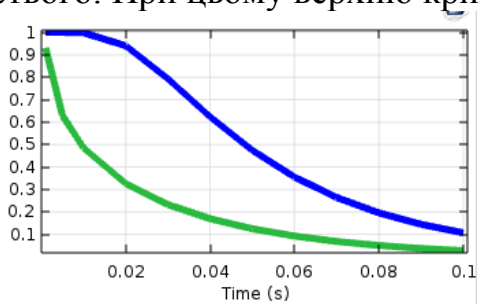
`Data` – **Time selection** = `All`

В секцію вибору введіть 1, клацнувши в графічній зоні по кубу. Далі введіть

`Expression` – **Expression** = `T`

LMB > Evaluate – Table 1 - Point Evaluation 1(T)

В таблиці **Table 1** буде додано стовпець зі значеннями середньої температури в кубі в різні моменти часу (див. попередній рисунок). Якщо клацнути мишею по кнопці Table Graph, яка розташована над таблицею, то будуть побудовані графіки ламаних, абсциса яких буде взята з першого стовпця таблиці, а ординати – з другого і третього. При цьому верхню криву ми будували раніше.



Існує інший спосіб побудови графіка залежності середньої температури куба від часу. Для цього потрібно створити джерело даних, яке буде представляти інтеграл від температури по об’єму куба (або середнє по об’єму).

Results – Data Sets: RMB > More Data Sets - Integral

В ПВ створеного вузла **Integral 1** введіть:

Data – **Data set** = Study 1/Solution 1

Settings – **Geometry level** = Volume

В графічну групу **1D Plot Group 4** додайте і налаштуйте графічний вузол типу Global. Для цього виконайте інструкції

1D Plot Group 4-> Global

В ПВ створеного вузла **Global 1** введіть:

Data – **Data set** = Integral 1

Data – **Time selection** = All

y-Axis Data – **Expression** = T (*введення виконується в таблиці*)

x-Axis Data – **Parameter** = Time

Виконайте команду Plot і отримайте графік такий же, як і попередній, що побудований по табличним даним.

Побудуємо графік середньої температури по прерізу куба $x = 0.9$. Спочатку треба створити джерело даних для цього перерізу.

Results->**Data Set**->Cut Plane

В ПВ створеного вузла **Cut Plane 1** введіть:

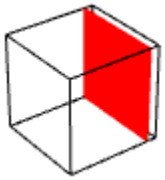
Data – **Data set** = Study 1/Solution 1

Plane Data – **Plane type** = Quick

Plane Data – **Plane:** = yz-planes

Plane Data – **x-coordinate:** = 0.9

Побудуйте зображення цього джерела даних командою Plot.



Створіть джерело даних, яке буде представляти середнє значення температури по площині перерізу. Для цього виконайте інструкції

Results – Data Sets: RMB > More Data Sets - Average

В ПВ створеного вузла **Average 1** введіть:

Data – **Data set** = Cut Plane 1

Settings – **Geometry level** = Take from data set

В графічну групу **1D Plot Group 4** додайте і налаштуйте графічний вузол типу Global. Для цього виконайте інструкції

1D Plot Group 4-> Global

В ПВ створеного вузла **Global 2** введіть:

Data – **Data set** = Average 1

Data – **Time selection** = All

y-Axis Data – **Expression** = T (*введення в таблиці*)

x-Axis Data – **Parameter** = Time

Щоб на графіку відобразити легенду, виконайте інструкції

1D Plot Group 4:

Legend – **Show legends, Position** = True, Upper right

1D Plot Group 4 - Point Graph 1:

Legends – **Show legends, Legends** = `True, 'Center'`

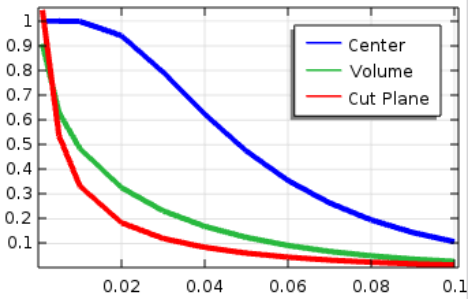
1D Plot Group 4 - Global 1:

Legends – **Show legends, Legends** = `True, 'Volume'`

1D Plot Group 4 - Global 2:

Legends – **Show legends, Legends** = `True, 'Cut Plane'`

Закінчіть побудову графіка командою Plot.



В динамічних задачах, так само як і при параметричному аналізі, зручно використовувати зонди. Для прикладу створимо два зонди. Перший `Domain Point` зонд буде відслідковувати температуру в центрі куба, другий – `Domain` зонд буде обчислювати середню температуру по об'єму.

Спочатку створимо точковий зонд.

Component 1 (comp1) – Definitions:

RMB > Probes – Domain Point Probe

Definitions - Domain Point Probe 1:

Point Selection – **Line entry method** = `None`

Point Selection – **Coordinates** – **x, y, z** = `0.5,0.5,0.5` (*три поля*)

Definitions - Domain Point Probe 1 - Point Probe Expression 1 (ppb1):

Expression – **Expression** = `T`

LMB > Update Results

Тепер створимо «просторовий» зонд.

Component 1 (comp1) – Definitions:

RMB > Probes – Domain Probe

Definitions - Domain Probe 1 (dom1):

Probe Type – **Type** = `Average`

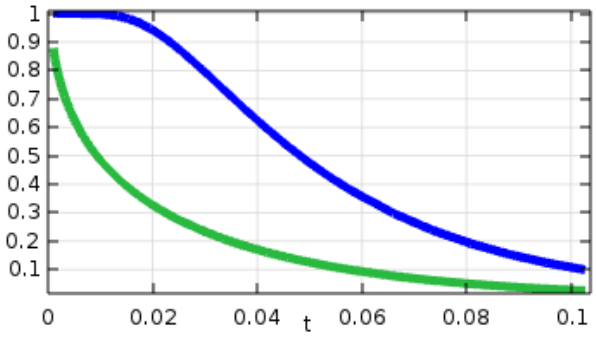
Source Selection – **Selection** = `All domains`

Expression – **Expression** = `T`

Study 1: RMB > Compute

В результаті переобчислення моделі будуть створені всі «зондові» вузли, побудована таблиця залежності зондових змінних від часу, і в графічній групі **Probe Plot Group 7** буде створено дві криві, такі ж самі, як раніше.

t	Temperature, Po	Temperature, D
0.0051887	1.00000	0.61987
0.0062042	0.99995	0.58466
0.0072196	0.99978	0.55376
0.0082351	0.99950	0.52623
0.0092505	0.99906	
0.010266	0.99831	
0.080738	0.19173	0.049451
0.088049	0.15382	0.039663
0.095361	0.12337	0.031815
0.10267	0.098948	0.025520



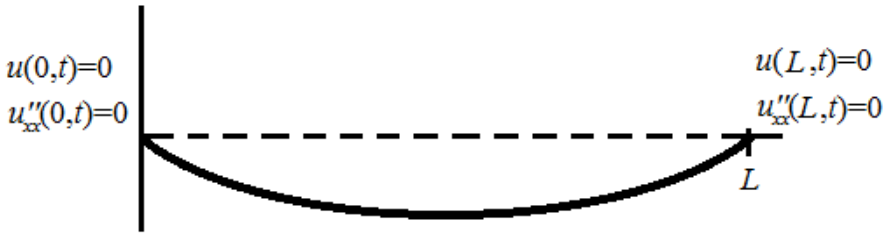
1.6.2. Коливання вільно опертої балки.

Дослідіть малі коливання тонкої балки довжиною L, кінці якої вільно оперті.

Урахування факту, що балка чинить опір згину, призводить до рівняння четвертого порядку

$$u''_{tt} = -\alpha^2 u''''_{xxxx}, \quad 0 < x < L, 0 < t < \infty, \tag{1}$$

де $\alpha^2 = K/\rho$, K - згинна жорсткість, ρ – лінійна густина балки (маса/од. довжини), $u(x,t)$ - вертикальне зміщення точки x балки в момент часу t.



Зрозуміло, що на кінцях балки виконуються граничні умови

$$u(0,t) = 0, u(L,t) = 0 \quad (0 \le t < \infty). \tag{2}$$

Використовуючи теорію тонких балок, можна показати, що згинальний момент в балці пропорційний величині u''_{xx}. Оскільки згинальні момент на кінцях вільно опертої балки дорівнює нулю, то отримуємо додаткові граничні умови

$$u''_{xx}(0,t) = 0, u''_{xx}(L,t) = 0 \quad (0 \le t < \infty). \tag{3}$$

В результаті, коливання балки описуються рівнянням (1), граничними умовами (2), (3), і початковими умовами

$$u(x,0) = \varphi(x), u'_t(x,0) = \psi(x), 0 \le x \le L.$$

Вивчимо коливання балки одиничної довжини $L = 1$, що вільно обперта на кінцях. Нехай $\alpha^2 = 1$ і початкові умови мають вигляд

$$\begin{aligned} u(x,0) &= \sin \pi x + 0.5 \sin 3\pi x \\ u'_t(x,0) &= 0 \end{aligned} \tag{4}$$

Аналітичний розв’язок цієї задачі відомий

$$w(x,t) = \cos(\pi^2 t) \sin \pi x + 0.5 \cos(9\pi^2 t) \sin(3\pi x). \tag{5}$$

Побудуємо чисельний розв'язок задачі за допомогою COMSOL і порівняємо його з (5).

COMSOL вміє розв’язувати системи ДРЧП не вище другого порядку. Тому задача повинна бути переписана в термінах системи двох ДР другого порядку. Якщо ввести позначення $v = u''_{xx}$, то рівняння (1) набуде вигляду $u''_{tt} = -v''_{xx}$ (при $\alpha^2 = 1$). В результаті задача може бути переписана у вигляді наступної системи

$$\begin{cases} u''_{tt} + v''_{xx} = 0 \\ -u''_{xx} + v = 0 \end{cases} \tag{6}$$

з граничними умовами

$$u(0,t) = 0,\ u(1,t) = 0,\ v(0,t) = 0,\ v(1,t) = 0,$$

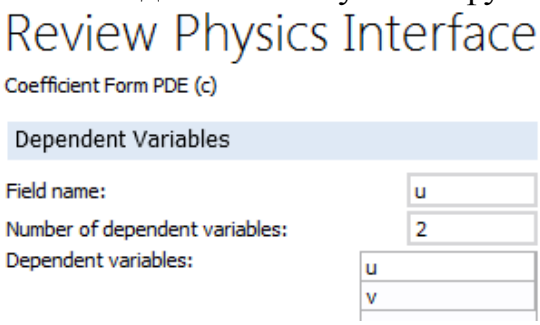
і початковими умовами (4).

Виконайте послідовність команд:

File->New->Model Wizard->1D->

Mathematics – PDE Interfaces – Coefficient From PDE(c)

Після натискання на кнопку Add в поле **Added physics interfaces** додається назва моделі. Справа в панелі `Review Physics Interface` в розділі **`Dependent Variable`** в поле **Number of dependent variables** введіть `2`, а в поле **Dependent variables** введіть імена шуканих функцій `u` і `v`.



Натисніть кнопку Study. У відкрившомуся вікні `Select Study` виберіть варіант `TimeDependent`. Завершіть створення дерева моделі натисканням на кнопку Done.

Відкрийте ПВ вузла **Component 1 (comp 1)** і введіть

`General` - **Unit system** = `None`

Геометрія моделі складається з одного відрізка. Побудуйте його.

`Geometry`**->`Pimitives`->** Interval

За замовчуванням параметри відрізка збігаються з тими, які нам потрібні. Не міняйте їх. Завершіть креслення геометрії клацанням по кнопці Build All Objects.

Побудуйте СкЕ сітку. Для цього в ПВ вузла **Mesh 1** введіть:

`Mesh Settings` –**Element size** = `Finer`

і натисніть кнопку Build All. В графічному вікні ви побачите зображення відрізка з точками розбиття.

Відкрийте ПВ вузла **Coefficient Form PDE (c) - Coefficient Form PDE 1**. В розділі **`Equation`** наведено загальний вигляд коефіцієнтної форми системи ДРЧП.

$$e_a \frac{\partial^2 \boldsymbol{u}}{\partial t^2} + d_a \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \left(-c \nabla \boldsymbol{u} - \alpha \boldsymbol{u} + \gamma \right) + \beta \cdot \nabla \boldsymbol{u} + a \boldsymbol{u} = f \ ,$$

де $\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix}$ - шукана вектор - функція і $\nabla = \frac{\partial}{\partial x}$.

Система рівнянь (6) в матричному запису має вигляд

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \frac{\partial^2}{\partial t^2} \begin{pmatrix} u \\ v \end{pmatrix} + \frac{\partial}{\partial x} \left(- \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} u \\ v \end{pmatrix} \right) + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Щоб коефіцієнтна форма набула такого вигляду, необхідно ввести наступні значення:

Diffusion Coefficient $- \mathbf{c} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

Absorption Coefficient $- \mathbf{a} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

Source Term $- \mathbf{f} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Mass Coefficient $- \mathbf{e_a} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

Тензори d_a, α, β і вектор γ в нашій задачі слід покласти рівними нулю. Перевірте також, що в поточній панелі (`Coefficient Form PDE 1`) в розділі **Domain Selection** стоїть 1, тобто обраний відрізок.

Задайте граничні умови, виконавши команду

`Physics->`**Boundaries**`->`Dirichlet Boundary Condition

Активуйте створений вузол **Dirichlet Boundary Condition 1** і натисніть кнопку Active в розділі **Boundary Selection**, якщо вона ще не натиснута. Мишкою клацніть по лівому кінцю відрізка (точка 1), а потім по правому кінцю (точка 2), додавши номери точок в секці вибору **Selection**. В розділі **Dirichlet Boundary Condition** в полях $\mathbf{r_1}$ і $\mathbf{r_2}$ введіть 0 - граничні значення функцій `u` і `v`.

Зайдіть в ПВ вузла **Initial Values 1** і перевірте, що в списку **Domain Selection** - **Selection** стоїть одиниця (номер відрізка). Потім введіть

Initial Values – **Initial value for u** – **u** =
`sin(pi*x)+0.5*sin(3*pi*x)`

Initial Values – **Initial value for v** – **v** =
`-pi^2*sin(pi*x)-0.5*9*pi^2*sin(3*pi*x)`

Initial Values – **Initial time derivative of u** – $\frac{\partial u}{\partial t}$ = 0

Initial Values – **Initial time derivative of v** – $\frac{\partial v}{\partial t}$ = 0

Зверніть увагу, що потрібно задати початкові умови для функції v, хоча за змістом задачі вони не потрібні. Оскільки $v = u''_{xx}$, то записати початкові умови

для функції v нескладно: $v(0,x)=u''_{xx}(0,x)$. Однак для функції v можна залишити нульові початкові значення або ввести будь-які інші. Результат не зміниться, оскільки початкові умови для функції v при розв'язанні задачі не використовуються.

Налаштуйте часовий інтервал і похибки обчислень. Для цього в ПВ вузла **Step 1: Time Dependent** введіть

`Study Settings` – **Times** = `range(0,0.05,0.5)`

Встановіть перемикач **`Study Settings`** - **Relative tolerance** і задайте відносну похибку обчислень `0.00001`.

Виконайте команду

`Study`-> Show Default Solver

відобразивши підвузли гілки **Study 1** дерева моделі (якщо в дереві їх ще немає). Розкрийте гілку **Study 1** і активуйте вузол **Study 1 - Solver Configurations - Solution 1 - Time - Dependent Solver 1**. В його ПВ введіть

`Absolute Tolerance` – **Tolerance** = `0.00001`

Виконайте обчислення командою

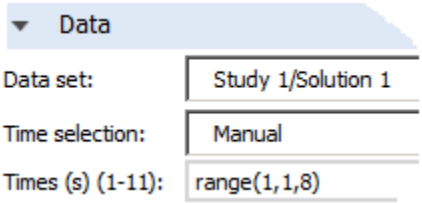
`Home`->Compute

В гілці **Results - Data Sets** з'явиться гілка **1D Plot Group 1** групи графіків з вузлом **Line Graph 1**, що містить графіки розв'язку $u(t,x)$ в різні моменти часу. В ПВ вузла **1D Plot Group 1** ведіть

`Data` – **Time selection** = `Manual`

`Data` – **Time(s)(1-11)** = `range(1,1,8)`

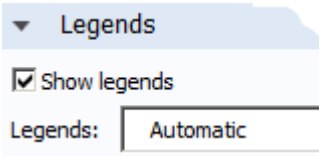
вибравши цим перші 8 моментів часу, щоб побудувати графіки тільки для них.



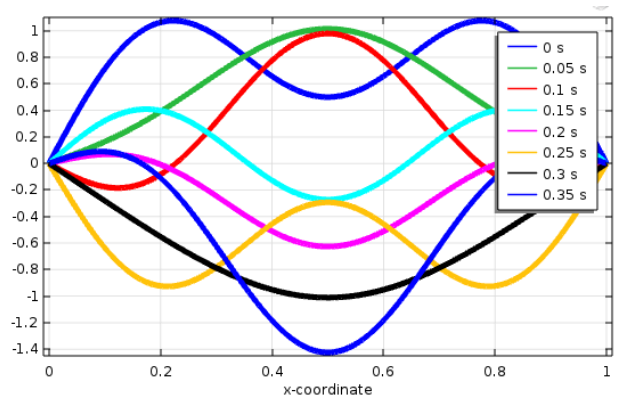
В ПВ вузла **1D Plot Group 1 - Line Graph 1** введіть

`x-Axis Data` – **Parameter, Expression** = `Expression, x`

щоб віднести графіки до координати x вздовж осі балки. Для відображення легенди в ПВ вузла **1D Plot Group 1 - Line Graph 1** встановіть перемикач **Show legends**.



В результаті ви отримаєте графіки нейтральної вісі балки для моментів часу, наведених в «легенді».



Для порівняння побудуємо графіки нейтральної вісі, використовуючи чисельний розв’язок і аналітичну формулу (5).

Створіть групу одновимірних графіків командою

`Results` -> 1D Plot Group

В ПВ створеного вузла **1D Plot Group 2** введіть:

`Data` – **Data set** = `Study 1/Solution 1`

`Data` – **Time selection** = `Manual`

`Data` – **Time(s)**(`1-11`) = `1,4,8`

Потім двічі виконайте команду

`1D Plot Group 2`-> Line Graph

В ПВ створеного вузла **Line Graph 1** в розділі **`Selection`** оберіть відрізок (його номер 1 з'явиться в секції вибору) і введіть

`y-Axis Data` – **Expression** = `u`

`x-Axis Data` – **Parameter, Expression** = `Expression, x`

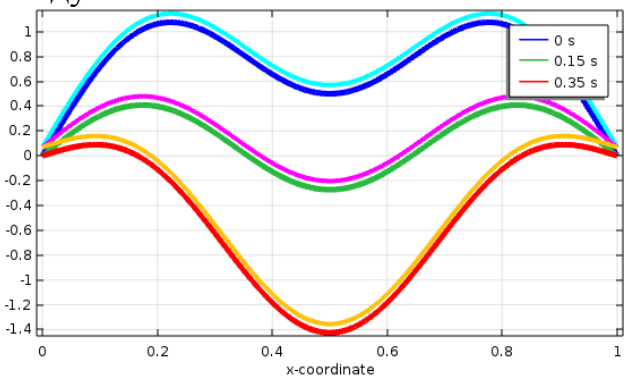
`Legends` – **Show legends** = `Включити`

В ПВ створеного вузла **Line Graph 2** в розділі **`Selection`** виберіть відрізок і введіть

`y-Axis Data` – **Expression** = `cos(pi^2*t)*sin(pi*x)+`
`0.5*cos(9*pi^2*t)*sin(3*pi*x)+0.07`

`x-Axis Data` – **Parameter, Expression** = `Expression, x`

В результаті графічна група буде містити графіки нейтральної вісі, побудовані для трьох моментів часу (`0`, `0.15` і `0.35`) з використанням чисельного розв’язку та аналітичної формули (5). Для наочності графіки аналітичного розв’язку зсунуті вгору на невелику константу `0.07`. Якщо її з формули забрати, то криві співпвдуть.



Вправа. Побудуйте графік різниці чисельного і аналітичного розв'язків. Знайдіть абсолютну похибку (максимум модуля різниці цих розв'язків). Побудуйте графік різниці перших похідних по x чисельного і аналітичного розв'язків. Побудуйте і порівняйте графіки функції $v(x,t)$, $u_{xx}(x,t)$ та $w_{xx}(x,t)$.

1.6.3. Осесиметричні коливання кругової мембрани

Вільні осесиметричні коливання кругової мембрани радіуса R, яка закріплена на контурі, описуються рівняннями

$$\frac{1}{a^2}\frac{\partial^2 u}{\partial t^2}=\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r},\quad u(t,R)=0 \tag{1}$$

Розв'яжемо задачу (1) з початковими умовами $u(0,r)=R^2-r^2$, $u'_t(0,r)=0$, які описують коливання з параболоїдальною початковою формою і нульовою початковою швидкістю. Для прикладу візьмемо $R=1$ і $a=1$.

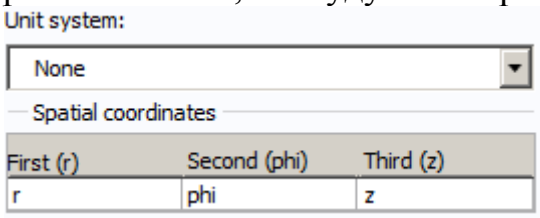
Створіть модель послідовністю команд:

File->New->Model Wizard->1D Axisymmetric->
Mathematics – PDE Interfaces – Coefficient From PDE(c) ->
TimeDependent

Відкрийте ПВ вузла **Component 1 (comp 1)** і введіть

General - **Unit system** = None

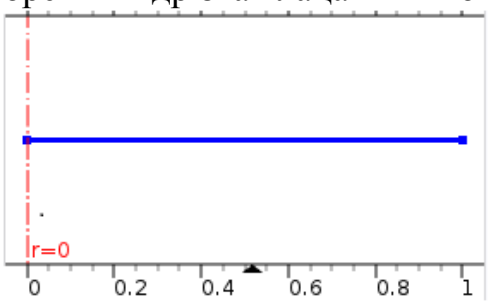
Тут же в таблиці розділу **Spatial coordinates** ви можете дізнатися (або поміняти) імена просторових змінних, які будуть використовуватися в моделі.



Геометрія складається з одного відрізка, обертанням якого створюється осесиметрична модель. Побудуйте його.

Geometry->**Pimitives->** Interval

За замовчуванням параметри відрізка збігаються з тими, які нам потрібні. Не міняйте їх. Завершіть створення відрізка клацанням по кнопці Build All Objects.



Відкрийте ПВ вузла **Coefficient Form PDE (c) - Coefficient Form PDE 1**. В розділі **Equation** цієї панелі наведено загальний вигляд коефіцієнтної форми ДРЧП

$$e_a\frac{\partial^2 u}{\partial t^2}+d_a\frac{\partial u}{\partial t}+\nabla\cdot(-c\nabla u-\alpha u+\gamma)+\beta\cdot\nabla u+\mathrm{a}\,u=f\,,$$

де $\nabla = \frac{\partial}{\partial r}$. Щоб це рівняння прийняло вид (1), необхідно ввести наступні значення:

Diffusion Coefficient – **c** = 1

Absorption Coefficient – **a** = 0

Source Term – **f** = 0

Mass Coefficient – $\mathbf{e_a}$ = 1

Damping or Mass Coefficient – $\mathbf{d_a}$ = 0

Convection Coefficient – $\boldsymbol{\beta}$ = -1/r

Значення коефіцієнтів α і γ залиште без змін (нульовими). Перевірте також, що в поточній панелі (Coefficient Form PDE 1) в секції вибору розділу **Domain Selection** стоїть 1, тобто обраний відрізок.

Задайте граничні умови, виконавши команду

Physics->**Boundaries**->Dirichlet Boundary Condition

Активуйте створений вузол **Dirichlet Boundary Condition 1**. В його ПВ в розділі **Boundary Selection** натисніть кнопку Active (якщо вона ще не активна) і мишею виділіть праву точку відрізка. Її номер 2 буде додано в секцію вибору **Selection**. В розділі **Dirichlet Boundary Condition** в полі **r** залиште значення 0.

Відкрийте в ПВ вузла **Zero Flux 1** і переконайтеся, що для лівої точки (номер 1) гранична умова не застосовується (насправді в цій точці воно автоматично задається умовою симетрії $\left.\frac{\partial u}{\partial r}\right|_{point1} = 0$).

Зайдіть в ПВ вузла **Initial Values 1** і перевірте, що в списку **Domain Selection** – **Selection** стоїть одиниця (індикатор відрізка). Потім введіть

Initial Values – **Initial value for u** – **u** = 1-r^2

Initial Values – **Initial time derivative of u** – $\frac{\partial u}{\partial t}$ = 0

Налаштуйте часовий інтервал і похибку обчислень. В ПВ вузла **Step 1: Time Dependent** введіть

Study Settings – **Times** = range(0,0.05,10)

Цим ми задаємо інтервал часу (від 0 до 10), на якому буде визначатися чисельний розв'язок, а також інтервал між моментами часу, для яких розв'язок буде зберігатися.

Встановіть прапорець в перемикачі **Study Settings** – **Relative tolerance** і задайте відносну похибку обчислень 0.00001.

Виконайте команду

Study-> Show Default Solver

відкривши підвузли гілки **Study 1** дерева моделі. Розкрийте цю гілку і активуйте вузол **Study 1 - Solver Configurations - Solution 1 - Time - Dependent Solver 1**. В його ПВ введіть

Absolute Tolerance – **Tolerance** = 0.00001

Побудуйте СкЕ сітку. Для цього в ПВ вузла **Mesh 1** введіть:

Mesh Settings –**Element size** = Finer

і натисніть кнопку Build All. У графічному вікні ви побачите зображення відрізка з точками розбиття.

Виконайте обчислення командою

Home->Compute

В гілці **Results - Data Sets** з'являться два вузли **Study 1/Solution 1** і **Revolution 1D 1**, які містять розв'язки, а також вузли, що містять їх графіки. Спочатку графічне вікно, пов'язане з вузлом **1D Plot Group 1** містить графіки розв'язку $u(t,r)$ для всіх моментів часу з масиву range(0,0.05,10). Його ми задали в ПВ вузла **Step 1: Time Dependent** в полі **Times**. Побудований графік малоінформативний, оскільки кривих багато. Виберемо з них тільки деякі. Для цього в ПВ вузла **1D Plot Group 1** введіть:

Data– **Data set** = Study 1/Solution 1

Data– **Time selection** = Manual

Data– **Time(s)**(1-101) = range(1,1,14)

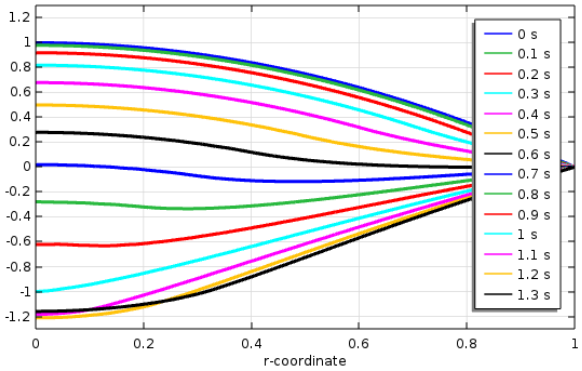
Axis–**Manual axis limits**= Включити

Axis–**x minimum, x maximum** = 0, 1 (*два поля*)

Axis–**y minimum, y maximum** = -1.5 , 1.5 (*два поля*)

Legend–**Show legends** = Включити

Задання в полі **Time(s)** списку номерів моментів часу дозволяє побудувати графіки тільки в бажані моменти. Вмикання «легенди» (**Show legends** = Включити) в графічній групі (а також в дочірньому графічному вузлі **Line Graph 1**) відображає на графіку таблицю відповідності моментів часу і кривих.



Графічна група **2D Plot Group 2** відображає графік розв'язку $u(t,x,y)$ для моменту часу, який обраний в списку **Data** – **Time(s)**. Встановимо для відображення нульовий момент часу.

Data – **Time (s)** = 0

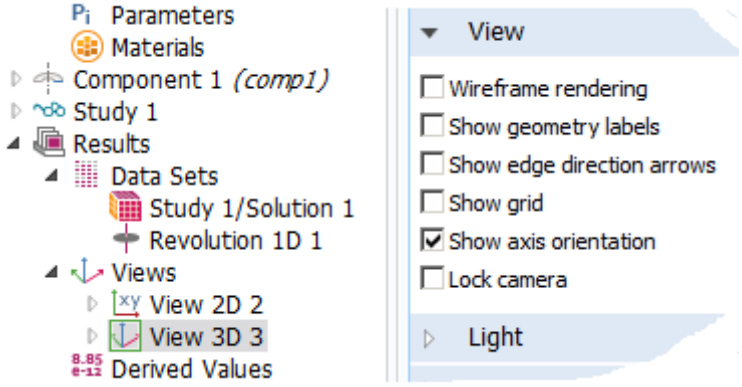
Скорегуємо джерело даних **Revolution 1D 1**, щоб воно представляло повне коло. Для цього в ПВ вузла **Data Sets - Revolution 1D 1** введіть

Revolution Layers – **Start angle, Revolution angle** = 0, 360

і натисніть кнопку Plot.

Клацніть правою кнопкою миші по гілці **2D Plot Group 2 - Surface 1** і додайте підвузол **Height Expression 1**. В його ПВ в полі **Scale factor** встановіть масштабний коефіцієнт вертикальних розмірів 1, і натисніть кнопку Plot. На

зображенні, крім поверхні, буде показана просторова координатна сітка. Щоб її видалити в ПВ вузла **Results - Views -View 3D 3** зніміть перемикач **Show grid**.



Включати і виключати координатну сітку можна також кнопкою Show Grid, яка розташована над графічним вікном.

Змінюючи в ПВ вузла **2D Plot Group 2** в списку `Data` – **Time(s)** значення моменту часу, ви будете отримувати новий графік поверхні мембрани. Однак кольорова палітра для кожного графіка буде своя. Щоб графіки в різні моменти часу використовували однакову палітру і розташовувалися в одній і тій же просторовій зоні, потрібно «вручну» задати діапазони зміни кольору і даних. Для цього в ПВ вузла **2D Plot Group 2 – Surface 1** введіть

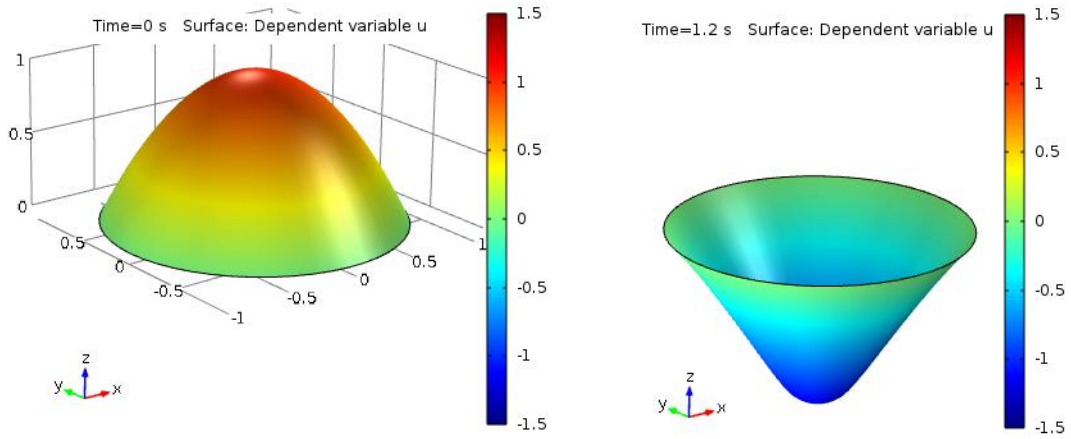
`Range` – **Manual color range** = `Включити`

`Range` – **Manual color range – Minimum, Maximum** = `-1.5, 1.5`

`Range` – **Manual data range** = `Включити`

`Range` – **Manual data range – Minimum, Maximum** = `-1.5,1.5`

Зображення поверхні мембрани в нульовий момент часу (з координатної сіткою), і в момент `1.2` (без сітки) показані на наступному рисунку. Щоб отримати такі зображення, можливо, вам буде потрібно змістити об'єкти в графічному вікні. Для цього підведіть покажчик миші до зображення, натисніть праву кнопку і, не відпускаючи, перемістіть мишу (а разом з нею зображення) в нове місце.



Побудуйте графік лінії перерізу поверхні мембрани вертикальною площиною. Для цього спочатку додайте відповідне джерело даних, виконавши команду

`Results`-> Cut Line 2D

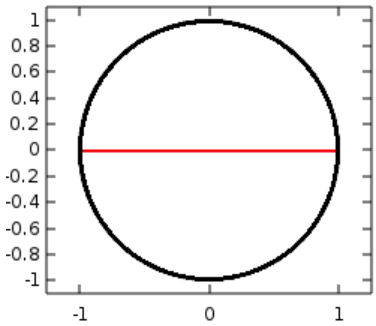
В ПВ створеного вузла **Cut Line 2D 1** введіть

`Data` – **Data set** = `Revolution 1D 1`

Line Data – **Point 1** – **r** = -1

Line Data – **Point 2** – **r** = 1

Побудуйте зображення джерела даних командою Plot.



Тепер створіть групу одновимірних графіків командою

Results -> 1D Plot Group

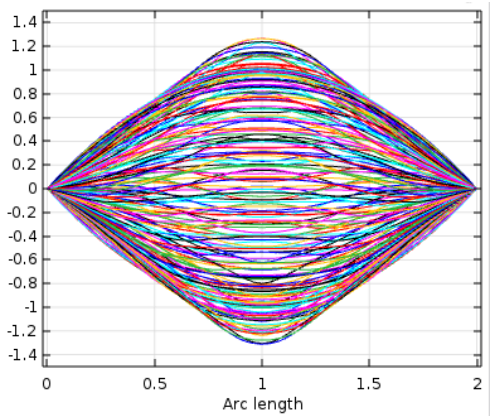
В ПВ створеного вузла **1D Plot Group 3** введіть:

Data – **Data set** = Cut Line 2D 1

Потім виконайте команду

1D Plot Group 3-> Line Graph

Налаштування ПВ створеного вузла **Line Graph 1** залиште без змін. В результаті графічна група буде містити велику кількість графіків для всіх моментів часу з масиву range(0,0.05,10).



Зверніть увагу, що горизонтальна вісь віднесена до довжини дуги джерела даних **Cut Line 2D 1**. Наведені графіки малоінформативні, але вони знадобляться для побудови анімації. Для цього виконайте команду

Results –> Animation –> Player

В ПВ створеного вузла **Animation 1** введіть:

Scene – **Subject** = 1D Plot Group 3

Target – **Target** = Player

Animation Editing – **Loop over** = Time

Animation Editing – **Time selection** = All

Frames – **Frame selection** = All

Згенеруйте кадри анімації, натиснувши кнопку Generate Frame. По завершенні процесу запустіть анімацію кнопкою Play, яка розташована над графічним вікном.

Збережіть поточний файл, наприклад, з ім'ям vibrmembrane.mph, бо він нам ще знадобиться.

Вправа. Розв'яжіть задачу про коливання квадратної мембрани $\Omega = [0,1]\times[0,1]$, яка закріплена на контурі. Для цього розгляньте рівняння $\frac{\partial^2 u}{\partial t^2} = a^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$ з граничною умовою $u|_{\partial\Omega} = 0$. Задайте початковий профіль у вигляді $u(0,x,y) = \sin\pi x \sin\pi y$, і нульову початкову швидкість. Порівняйте чисельний розв'язок з аналітичним, який має вигляд $u(t,x,y) = \cos\left(\sqrt{2}\,\pi a t\right)\sin\pi x \sin\pi y$, побудувавши графік різниці розв'язків уздовж діагоналі квадрата. Побудуйте анімацію коливань квадратної мембрани.

1.7. Математичні вирази

У попередніх прикладах ми кілька разів використовували математичні вирази. Коротко опишемо правила їх побудови.

Математичні операції позначаються стандартними значками.

Унарні операції: !, –, + (логічне заперечення, мінус, плюс).

Бінарні арифметичні операції: ^, *, /, +, – (піднесення до степеня, множення, ділення, додавання і віднімання).

Операції порівняння: <, <=, >, >=, ==, !=

Логічні операції: &&, || (логічне І , логічне АБО).

Пріоритет арифметичних операцій звичайний. Для зміни порядку виконання операцій використовуються круглі дужки.

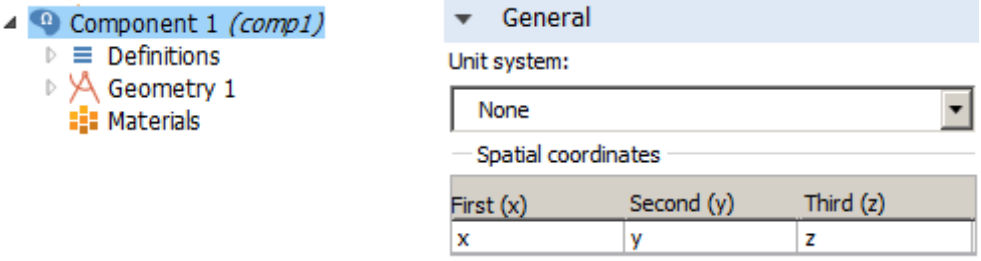
У виразах COMSOL Multiphysics можна використовувати різні вбудовані константи і функції. Їх імена зарезервовані і не можуть бути перепозначені користувачем. Якщо ви спробуєте використати таке ім'я для змінної, параметра або функції, то текст, який ви введете, стане помаранчевим (попередження) або червоним (помилка). При підведенні покажчика миші до такого тексту спливе пояснювальна підказка.

Ось приклади деяких зарезервованих імен:

- константа `pi` (3.141592...),
- `eps` – мінімальна відносна похибка (машинна іпсилон 2^{-52}, або приблизно $2.2204\cdot10^{-16}$);
- `inf`, `Inf` – нескінченність, ∞;
- `Nan`, `nan` (Not-a-number) – невизначене значення таке, як 0/0 або ∞/∞;
- фізичні сталі, наприклад, `g_const` (прискорення вільного падіння), `R_const` (універсальна газова стала)
- уявна одиниця `i` (або `j`) (тобто $\sqrt{-1}$);
- змінна часу `t`;
- назви математичних функцій, таких як `cos`, `sin`, `exp`, `log` та інших. Перелічимо позначення основних з них (імена аргументів можуть бути іншими): `abs(x)`, `acos(x)`, `asin(x)`, `cos(x)`, `cosh(x)`, `erf(x)`, `exp(x)`, `floor(x)`, `log(x)` (натуральний логарифм), `log10(x)`

(десятковий логарифм), `max(a,b)`, `min(a,b)`, `sign(u)` (функція знаку), `sin(x)`, `sinh(x)`, `sqrt(x)`, `tan(x)`, `tanh(x)`.

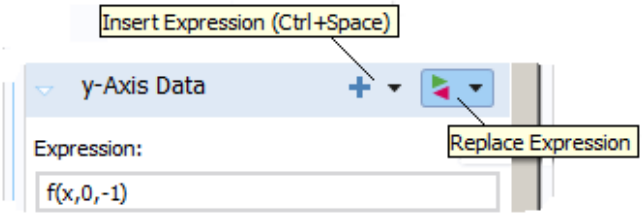
В моделях розмірності 1D, 2D, 3D використовуються зумовлені імена декартових координат `x,y,z`. Однак їх можна змінити в панелі властивостей гілки **Component**. В таблиці **`Spatial coordinates`** можна ввести інші позначення просторових змінних.



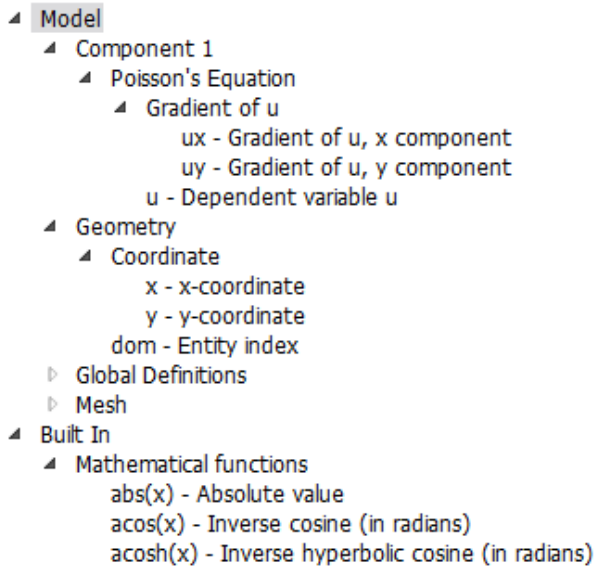
В осесиметричних моделях `1D Axisymmetric` і `2D Axisymmetric` циліндричні координати зазвичай позначаються `r, phi, z`.

У виразах часто використовуються частинні похідні. Якщо `u` позначає ім'я розв'язку, наприклад, двовимірного рівняння Пуассона, а `x` і `y` представляють декартові координати на площині, то у виразах для позначення перших і других похідних слід використовувати наступні ідентифікатори: `ux, uy, uxx, uyy, uxy`. Наприклад, ім'я `ux` позначає частинну похідну розв'язку `u` по просторовій координаті `x`, а `uxx` відповідає другий частинній похідній розв'язку по `x`. Якщо просторові координати мають інші імена, наприклад, `r` і `psi`, то тоді потрібно використовувати позначення `ur, upsi, urr, urpsi` і т.д. Коли розв'язок залежить від часу `t`, то можна використовувати частинні похідні по `t`, які позначаються наступним чином: `ut, uxt, uyt, utt, uxxt, uxyt, uyxt,uyyt, uxtt, uytt, uxxtt, uxytt, uyytt`. Ім'я змінної часу `t` зарезервовано і користувач не може його змінити.

Більшість позначень змінних, використовуваних в конкретній моделі, можна дізнатися з вікна, яке відкривається після клацання мишею по кнопках Replace Expression або Insert Expression, розташованих вгорі полів введення виразів.



Частина вмісту цього вікна показана на наступному рисунку.



Якщо вікно відкривалося після вибору кнопки Replace Expression, то подвійне клацання мишею по елементу цього вікна заміняє вміст поля **Expression**. Вибір елемента цього вікна після вибору кнопки Insert Expression вставляє відповідний текст в поточну позицію курсора в поле **Expression**.

Крім функцій, для моделювання та обчислювання результатів застосовуються спеціальні оператори. Опишемо деякі з них.

- `d(f,x)`, `pd(f,x)` – диференціювання виразу `f` по змінній `x`. Для похідних розв'язку `u` другого порядку можна використовувати різні позначення. Наприклад, `uxx`, `d(ux,x)` і `d(d(u,x),x)` еквівалентні всередині розрахункової області. На межі області `d(u,x)=0`, а `ux` дорівнює середньому по суміжним областям. Оператор `pd` діє аналогічно оператору `d`. Основна відмінність полягає в тому, що `pd(u,x)` дорівнює нулю, а не `ux`, тому, що в `pd` операторі `u` розглядається як незалежна змінна. Наприклад, вирази `d(u^2,u)` і `pd(u^2,u)` обидва повертають `2*u`. Вираз `d(u+x,x)` дорівнює `ux + 1`, а вираз `pd(u+x,x)` дорівнює 1, тому, що `u` розглядається як незалежна змінна. Вираз `d(u,t)` дорівнює `ut`, тоді як `pd(u,t)` дорівнює нулю. Вираз `d(u*v,t)` дорівнює `u*vt+ut*v`, а `pd(u*v,t)` дорівнює нулю.
- `if(condition, expr1, expr2)` – умовний вираз, повертає другий або третій аргумент в залежності від істинності аргументу `condition`. Наприклад, оператор `if(x==0,1,sin(x)/x)` являє неперервний вираз для всіх дійсних `x`, включаючи `x = 0`.
- Оператор `integrate(expr,var,lower,upper)` обчислює інтеграл від виразу `expr` по змінній `var` на відрізку `[lower,upper]`. Можна використовувати п'ятий параметр – відносну похибку обчислення. Вона за замовчуванням дорівнює $1 \cdot 10^{-3}$.
- Оператор підсумовування має наступний синтаксис:
  `sum(expr, index, lower, upper)`
  Наприклад, `sum(i^2,i,1,4)`$=1^2+2^2+3^2+4^2=30$.

- Оператори timeint і timeavg обчислюють інтеграл і інтегральне середнє за часом. Оператор timeint(t1,t2,expr) обчислює інтеграл виразу expr від моменту часу t1 до моменту t2. Інтеграл визначається чисельно. Оператор обчислювання середнього за часом має синтаксис timeavg(t1,t2,expr) і відрізняється від timeint тим, що результат інтегрування ділиться на t2-t1.
- Оператори timemax(t1,t2,expr) і timemin(t1,t2,expr) визначають максимальне і мінімальне значення виразу expr на відрізку часу $t_1 \le t \le t_2$. Ці оператори можна використовувати тільки під час обробки результатів обчислювання, тобто в полях вікон властивостей вузлів гілки **Results**. Якщо вираз expr крім змінної часу t залежить ще від просторових змінних, то максимум і мінімум шукається в кожній точці окремо і оператор повертає просторовозалежний вираз. Наприклад, якщо змінна u залежить від просторової координати r і часу t, то вираз v=timemax(t1,t2,u) повертає максимуми u по часу (t1≤t≤t2) в кожній точці r розрахункового інтервалу.
- Оператор with призначений для доступу до елементів розв'язку під час обробки результатів. В динамічних задачах він дозволяє отримати значення розв'язку на будь якому часовому кроці; при параметричному аналізі за його допомогою можна отримати доступ до розв'язку при фіксованому значенні параметра; в задачах на власні значення він дає доступ до власних функцій, які відповідають потрібним власним числам. Його першим аргументом є номер (моменту часу, параметра або власного числа). Другий аргумент являє ім'я розв'язку або вираз. Наприклад, в задачі на власні значення оператор with(3,u^2) представляє квадрат третьої власної функції.
- Оператор at(t0,sol) дає доступ до часозалежного розв'язку sol в будь який момент часу t0, викорстовуючи, якщо треба, інтерполяцію. Цей оператор можна використовувати при обробці результатів обчислення (в гілці **Results**) і неприпустимо використовувати при побудові моделі. Можна створити змінну моделі, яка містить цей оператор, але використовувати її треба лише на етапі постобробки.
- В системі існує кілька просторових операторів ati(координати,expr) з назвами at0(...), at1(...), at2(...) і at3(...). Вони обчислюють вираз expr в існуючій геометрії на *i* – вимірному примітиві в точці з заданими координатами. Оператор at0 використовується для нульвимірних примітивів (в точках), at1 – одновимірних (на ребрах), at2 – двовимірних (на гранях), at3 – тривимірних (в областях). Наприклад, at1(x_0,y_0,expr) обчислює вираз expr в двовимірній геометрії на ребрі в точці (x_0,y_0). Для коректного використання оператора ati повинен існувати примітив відповідної розмірності в точці з заданими координатами. Наприклад, at0 працює тільки у випадку, коли геометрія містить вершину в точці обчислювання. Майте на увазі, що кількість

координат точки, що передаються оператору `ati`, повинна співпадати з просторовою розмірністю геометрії моделі.

- Оператори `attimemax(t1,t2,expr1,expr2)` і аналогічний йому `attimemin(...)` шукають момент часу t_0 $(t_1 \le t_0 \le t_2)$, в який вираз `expr1` досягає максимуму (мінімуму), і в цей момент t_0 обчислюють вираз `expr2`. Аргументи `t1` і `t2` повинні бути константами, наприклад, вони не можуть обчислюватися за допомогою інших операторів. Оператори `attimemax` і `attimemin` можна використовувати лише на етапі обробки результатів обчислювання.
- Оператор `subst(expr,var1,expr1,...)` робить підстановку. Першим аргументом є вираз `expr`, в якому буде виконуватися заміна. Далі ідуть пари аргументів, кожна з яких складається з імені змінної і значення, яке замість неї слід підставити в початковий вираз. Наприклад, `subst(x^2-y^2,x,0.5,y,0.3)`.

Існує велика кількість інших операторів. З деякими ми познайомимо вас пізніше.

Продемонструємо використання операторів на прикладах. Розглянемо тестову модель і поставимо до неї кілька питань. Хоча більшість з них штучні, вони продемонструють техніку застосування деяких операторів.

Приклад 1. Розв'яжемо рівняння Пуасона в одиничному квадраті $[0,1]\times[0,1]$

$$\Delta u = 16\cdot\left(x^2+y^2\right),$$

з граничними умовами Діріхлє:

$$u(x,0)=x^4,\ u(0,y)=y^4,\ u(x,1)=\left(1+x^2\right)^2,\ u(1,y)=\left(1+y^2\right)^2.$$

Після стандартного розв'язання буде поставлено кілька додаткових питань. Щоб на них відповісти, ми створимо змінні моделі, при обчисленні яких будуть використовуватися оператори.

Створіть модель послідовністю команд:

File->New->Model Wizard->2D->
Mathematics – Classical PDEs - Poisson's Equation(poeq)
-> Stationary

В ПВ кореневого вузла **Untitled.mph (root)** введіть:

`Unit system` - < > = `None`

Створіть функцію $f(x,y)$, яка буде удавати праву частину рівняння Пуасона.

`Home` **-> `Definitions`** -> Functions -> Local -> Analytic

В ПВ створеного вузла **Component 1 – Definitions – Analytic 1** введіть:

Function name = `f`

`Definition` – **Expression**= `16*(x^2+y^2)`

`Definition` – **Arguments**= `x,y`

Для контроля побудуйте графік функції, клацнувши мишею по кнопці Plot.

Геметрія моделі складається з одиничного квадрата. Створіть його.

`Geometry`**->`Pimitives`->** Square

В ПВ створеного вузла **Square 1** залиште значення параметрів за замовчуванням і закінчіть побудову геометрії командою

Geometry – Square 1 (c1) : LMB > Build All Objects

На створеному квадраті побудуйте СкЕ сітку з параметрами за замовчуванням.

Component 1 – Mesh 1 : LMB > Build All

Налаштуйте параметри математичної моделі.

Poisson's Equation (poeq) - Poisson's Equation 1:

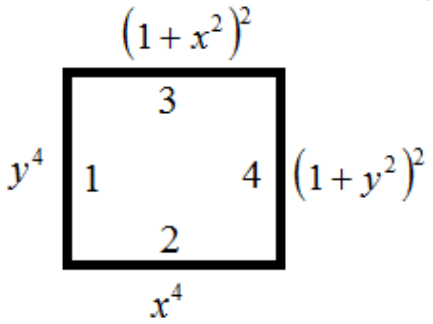
`Diffusion Coefficient` – **c** = `-1`

`Source Term` – **f** = `f(x,y)`

Задайте граничні умови. Оскільки вони на різних ділянках границі різні, то для кожної треба створити свій вузол `Dirichlet Boundary Condition`. Для цього 4 рази виконайте інструкцію:

`Physics` -> **`Boundaries`** – Dirichlet Boundary Condition

Потім у ПВ створених вузлів **Dirichlet Boundary Condition X** (де X=1,2,3,4) по черзі задайте граничні умови, які показано на наступному рисунку.



Наприклад, для лівої сторони квадрата потрібно виконати інструкції:

Dirichlet Boundary Condition 1:

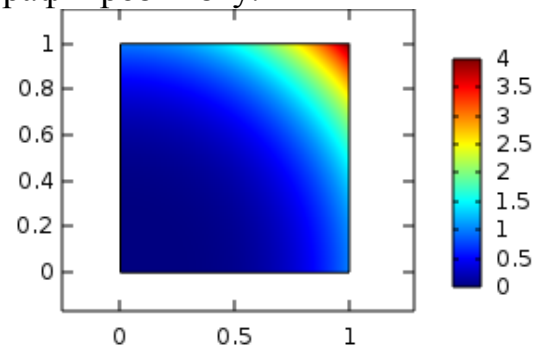
`Boundary Selection` – **Selection** = `1`

`Dirichlet Boundary Condition` – **r** = `y^4`

Після введення граничних умов на всіх чотирьох сторнах квадрата виконайте обчислення.

`Home` **->** Compute

В гілці **Results** з'явиться двовимірна графічна група **2D Plot Group 1** з вузлом **Surface 1**, що містить графік розв'язку.



Знайдемо значення розв'язку в окремих точках геометрії, наприклад, в одній внутрішній точці, в одній точці на контурі і в одній вершині. Для цього ми створимо кілька змінних моделі, які будуть містити потрібні значення (інший спосіб полягає в використання джерел даних `Cut Point 2D`).

Створимо вузол локальних змінних командою

`Home` **-> Definitions** – Variables – Local Variables

Заповніть перші 3 рядка таблиці так, як показано нижче (стовпець **Description** ми замінили більш докладним поясненням).

Component 1 – Definitions – Variables 1:

Name	**Expression**	**Пояснення**
`up0`	`at2(0.5,0.3,u)`	Значення розв'язку у внутрішній точці $u(0.5,0.3)$
`up1`	`at1(1,0.5,u)`	Значення в точці границі $u(1,0.5)$
`up2`	`at0(1,1,u)`	Значення в вершині $u(1,1)$

Обновіть обчислення для того, щоб нові змінні стали працездатними.

Study 1: RMB > Update Solution

Надрукуйте значення змінних `up0, up1, up2`, використовуючи вузли `Global Evaluation`.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 1:

`Expression` – **Expression** = `comp1.up0`

LMB > Evaluate

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 2:

`Expression` – **Expression** = `comp1.up1`

LMB > Evaluate – Table 1

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 3:

`Expression` – **Expression** = `comp1.up2`

LMB > Evaluate – Table 1

В таблиці `Table 1` будуть надруковані наступні значення змінних.

comp1.up0	comp1.up1	comp1.up2
0.11559	1.5625	4.0000

Припустимо, що нам потрібна змінна, яка містить суму значень розв'язку u в деяких точках, наприклад, розташованих на горизонтальному перерізі квадрата $y=0.5$ в точках $x=0.2,0.4,0.6,0.8$, тобто нас цікавить сума $\sum_{k=1}^{4} u(k/5,0.5)$. Суму можна обчислювати за допомогою оператора `sum(...)`, а значення розв'язку `u` у внутрішніх точках геометрії обчилюється оператором `at2(...)`.

Створимо змінну моделі `us`, додавши в таблицю змінних наступний рядок.

Name	**Expression**	**Пояснення**
`...`	`...`	...
`us`	`sum(at2(k/5,0.5,u),k,1,4)`	$\sum_{k=1}^{4} u(k/5,0.5)$

Обновіть розв'язок.

Study 1: RMB > Update Solution

Надрукуйте значення змінної `us`.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 4:

Expression – **Expression** = comp1.us

LMB > Evaluate – Table 1

В таблицю Table 1 буде додано стовпець us зі значенням 1.4164.

comp1.up0	comp1.up1	comp1.up2	comp1.us
0.11559	1.5625	4.0000	1.4164

Нехай нам потрібно значення інтеграла від розв'язку вздовж горизонтального перерізу квадрата $y = 0.5$, тобто нас цікавить інтеграл $\int_0^1 u(x,0.5)\,d\,x$. Його можна обчислити за допомогою оператора integrate(...).

Створимо змінну моделі ui, яка містить потрібне значення, додавши в таблицю змінних наступний рядок.

Name	**Expression**	**Пояснення**
...	...	...
ui	integrate(at2(s,0.5,u),s,0,1)	$\int_0^1 u(x,0.5)\,d\,x$

Обновіть розв'язок.

Study 1: RMB > Update Solution

Надрукуйте значення змінної ui.

Results -> **Derived Values** – Global Evaluation

Derived Values - Global Evaluation 4:

Expression – **Expression** = comp1.ui

LMB > Evaluate – Table 1

comp1.up0	comp1.up1	comp1.up2	comp1.us	ui
0.11559	1.5625	4.0000	1.4164	0.42916

Продемонструємо використання оператора підстановки subst(expr,...). Створимо 4 змінні моделі. Змінні p1 і p3 будуть містити вирази, в яких буде виконуватися заміна, а p2 і p4 – безпосередньо підстановки. Також створимо змінну $p5 = u\left(\sqrt{x}, \sqrt{y}\right)$ з «функціональною» заміною просторових координат.

Додайте в таблицю змінних наступні рядки.

Name	**Expression**	**Пояснення**
...	...	...
p1	x^2-y^2	
p2	subst(p1,x,0.5,y,0.8)	$p1(0.5,0.8)$
p3	x^0.25	$\sqrt[4]{x}$
p4	subst(p3,x,u)	$\sqrt[4]{u}$
p5	at2(sqrt(x),sqrt(y),u)	$u\left(\sqrt{x}, \sqrt{y}\right)$

Обновіть розв'язок.

Study 1: RMB > Update Solution

Для обчислення змінної p2 вчиніть як і раніше, створивши вузол Global Evaluation 6 (зробіть це самостійно).

comp1.up0	comp1.up1	comp1.up2	comp1.us	ui	p2
0.11559	1.5625	4.0000	1.4164	0.42916	-0.39000

Для перевірки працездатності змінної `p4` побудуємо її графік. Створіть двовимірну графічну групу з вузлом **Surface 1**, що містить графік функціонального забарвлення цієї змінної.

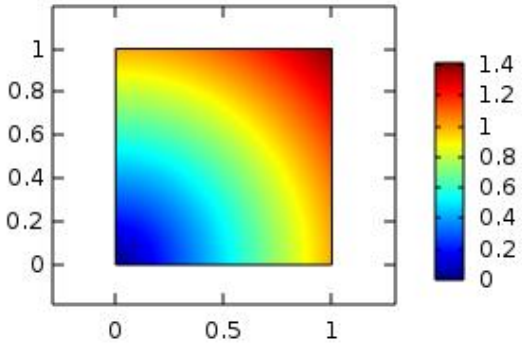
`Results->`2D Plot Group

`2D Plot Group 2` **`->Add Plot`** – Surface

2D Plot Group 2 – Surface 1:

`Expression` – **Expression** = `p4`

LMB > Plot



Аналогічно перевіримо працездатність змінної `p5`, побудувавши її графік.

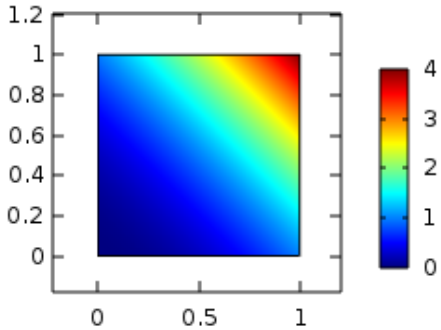
`Results->`2D Plot Group

`2D Plot Group 3` **`->Add Plot`** – Surface

2D Plot Group 3 – Surface 1:

`Expression` – **Expression** = `p5`

LMB > Plot



Створимо ще 2 змінні моделі, додавши в таблицю наступні рядки.

Name	**Expression**	**Пояснення**
`...`	`...`	...
`p6`	`at2(x,0.75,u)`	$u(x,0.75)$
`p7`	`at2(x,0.7,d(u,x))/2`	$\frac{\partial u}{\partial x}(x,0.7)$

Обновіть розв’язок.

Study 1: RMB > Update Solution

Для перевірки працездатності змінної `p6` побудуємо її графік, зважаючи на те, що вона залежить від однієї змінної x і отримана на горизонтальному перерізі квадрата $y = 0.75$. Створіть джерело даних, виконавши команду

`Results->` More Data Sets – Grid 1D

Джерела даних типу `Grid` використовуються для обчислення функцій та змінних тоді, коли немає потреби прив'язуватися до СкЕ сітки. У ДД `Grid 1D` розрахунковою областю є інтервал.

Data Sets – Grid 1D 1:

`Data` – **Source**= `Data set`

`Data` – **Data set** = `Study 1/Solution 1`

Інші параметри в ПВ створеного вузла **Data Sets – Grid 1D 1** залиште незмінними. Зокрема в полі **`Parameter Bounds`** – **Name** залишиться ім'я координати *x* вздовж відрізка [0,1].

Створіть одновимірну графічну групу, і в ній лінійний графік змінної `p6`.

`Results` -> 1D Plot Group

1D Plot Group 4 :

`Data` – **Data set** = `Grid 1D 1`

`Legends` – **Show legends** = `Включити`

`Legends` – **Position** = `Upper left`

`1D Plot Group 4`-> Line Graph

1D Plot Group 4 - Line Graph 1 :

`y-Axis Data` – **Expression** = `p6`

`x-Axis Data` – **Parameter, Expression** = `Expression,x`

`Legends` – **Show legends** = `Включити`

LMB > Plot (*крива* 1 *на наступному рисунку*)

Перевірте працездатність змінної `p7` (половина похідної вздовж горизонтального перерізу квадрата $y = 0.7$), розуміючи, що вона також залежить від однієї змінної *x*. Для цього додайте в графічну групу **1D Plot Group 4** лінійний графік змінної `p7`.

`1D Plot Group 4`-> Line Graph

1D Plot Group 4 - Line Graph 2 :

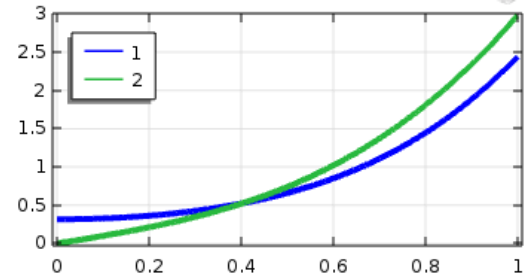
`y-Axis Data` – **Expression** = `p7`

`x-Axis Data` – **Parameter, Expression** = `Expression, x`

`Legends` – **Show legends, Legends** = `True, Manual`

`Legends` – **Legends** = `2` (*введення в таблиці легенд*)

LMB > Plot (*крива* 2 *на наступному рисунку*)



Остаточна таблиця змінних зображена на наступному рисунку.

Variables

Name	Expression
up0	at2(0.5,0.3,u)
up1	at1(1,0.5,u)
up2	at0(1,1,u)
us	sum(at2(k/5,0.5,u),k,1,4)
ui	integrate(at2(s,0.5,u),s,0,1)
p1	x^2-y^2
p2	subst(p1,x,0.5,y,0.8)
p3	x^0.25
p4	subst(p3,x,u)
p5	at2(sqrt(x),sqrt(y),u)
p6	at2(x,0.75,u)
p7	at2(x,0.7,d(u,x))/2

Зауваження. Окільки розв'язок задачі відомий $u=\left(x^2+y^2\right)^2$, то ви можете перевірити всі обчислення, створивши глобальну функцію з цим виразом, і виконати з нею всі попердні розрахунки.

■

Перевірку працездатності операторів, пов'язаних зі змінною часу, ми виконаємо в моделі коливання кругової мембрани, яка закріплена на контурі, має параболоїдальну початкову форму і нульову початкову швидкістю. Вона розглядалася в пункті **1.6.3** і була збережена в файл `vibrmembrane.mph`. Відкрийте цей файл і перезбережіть з іншим ім'ям, бо ми будемо вносити в нього багато змін.

Приклад 2. Відкрийте файл `vibrmembrane.mph`. Нагадаємо, що в ньому побудована одновимірна вісесиметрична (`1D Axisimmetric`) динамічна (`Time Dependent`) модель. Координата вздовж відрізка, який представляє геометрію, позначається літерою `r`. Це вказано в ПВ вузла **Component 1 (comp 1)** в полях таблиці **`Spatial coordinates`**.

Spatial coordinates

First (r)	Second (phi)	Third (z)
r	phi	z

Більшість операторів, які будуть розглядатися в цьому прикладі, можна використовувати лише на етапі обробки результатів обчислень, тобто в виразах панелей властивостей вузлів гілки **Results**. Можна створити змінні моделі, які містять такі оператори, але тоді їх можна буде використовувати лише на етапі постобробки. Тому ми відразу будемо використовувати оператори в підвузлах гілки **Results**.

Побудуємо графік розв'язку в монет часу $t=3$, використовуючи оператор `at(...)`. Виконайте наступні інструкції.

`Results` **`-> Plot Group ->`**2D Plot Group

2D Plot Group 4: `2D Plot Group 4->` Surface

2D Plot Group 4 – Surface 1:

`Data` – **Data set** = `Revolution 1D 1`

`Expression` – **Expression** = `at(3,u)`

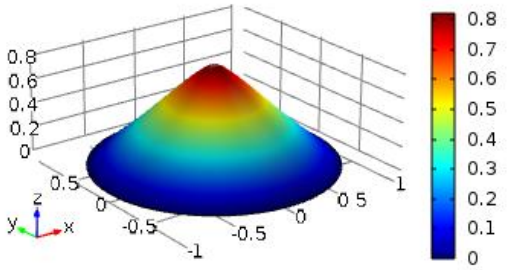
RMB> Height Expression

2D Plot Group 4 – Surface 1 – Height Expression 1:

`Axis` – **Scale factor** = `True, 1`

LMB>Plot

Буде побудований наступний профіль мембрани для моменту часу `t=3`.



Зверніть увагу, що в операторі `at(3,u)` перший аргумент позначає момент часу, в який розглядається розв'язок `u`. Схожим чином працює оператор `with(nummoment,expr)`, але першим агументом в динамічних задачах він приймає номер моменту часу, а не саме значення часу.

Побудуємо графік розв'язку на 52-му часовому кроці, використовуючи оператор `with (...)`. Виконайте наступні інструкції.

`Results` **`-> Plot Group ->`**2D Plot Group

Results - 2D Plot Group 5:

`Data` – **Data set** = `Revolution 1D 1`

`2D Plot Group 5->` Surface

2D Plot Group 5 – Surface 1:

`Expression` – **Expression** = `with(52,u)`

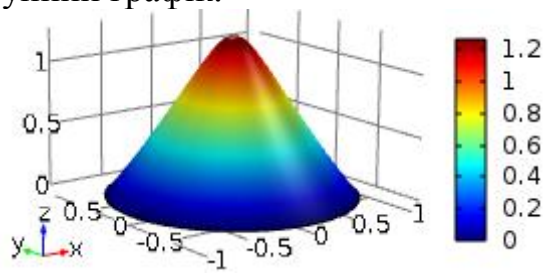
RMB> Height Expression

2D Plot Group 5 – Surface 1 – Height Expression 1:

`Axis` – **Scale factor** = `True, 1`

LMB>Plot

Буде побудований наступний графік.



Щоб дізнатися значення цього 52-го кроку (тобто моменту часу) теж можна застосувати оператор `with(...)`.

`Results ->` **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 1:

`Expression` – **Expression** = `with(52,t)`

LMB > Evaluate

Буде створена таблиця `Table 1` з двома стовпцями: моментами часу і обчисленим значенням `5.1` (для кожного моменту часу).

Time (s)	with(52,t)
0.0000	5.1000
0.10000	5.1000
0.20000	5.1000
0.30000	5.1000
0.40000	5.1000
. . . . .	
9.9000	5.1000
10.000	5.1000

Тобто 52-му кроку відповідає $t = 5.1$.

Судячи з графіків, на кожному часовому кроці максимальне зміщення досягається в центрі мембрани. Знайдемо момент часу, в який це зміщення максимальне. В системі існує оператор `attimemax(t1,t2,expr1,expr2)`. Він обчислює момент часу t_0 $(t_1 \le t_0 \le t_2)$, в який вираз `expr1` досягає максимуму, і в цей момент t_0 обчислює вираз `expr2`. Вираз `at1(0,u)` обчислює зміщення центру мембрани, тобто $u(0,t)$. Нагадаємо, що ми маємо одновимірну вісесиметричну (`1D Axisimmetric`) модель. Тому обчислення значення розв'язку `u` у внутрішній точці геометрії потрібно обчислювати за допомогою одновимірного опертора `at1`. Вираз `attimemax(0,10,at1(0,u),t)` обчислить значення t_0 $(0 \le t_0 \le 10)$, в який $u(0,t)$ досягає максимуму. Надрукуємо його.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 2:

`Expression` – **Expression** = `attimemax(0,10,at1(0,u),t)`

LMB > Evaluate – Table 1

В таблицю `Table 1` буде додано стовпець зі значеннями `5.1117`.

Time (s)	with(52,t)	attimemax(0,10,at1(0,u),t)
0.0000	5.1000	5.1117
0.10000	5.1000	5.1117
0.20000	5.1000	5.1117
0.30000	5.1000	5.1117

У оператора `attimemax(...)` є опція `'nointerp'`, яка при пошуку максимуму відключає інтерполяцію між часовими кроками. Замініть у вузлі **Global Evaluation 2** вираз і виконайте обчислення.

`Expression` – **Expression** =

`attimemax(0,10,at1(0,u),t, 'nointerp')`

LMB > Evaluate – Table 1

В таблицю `Table 1` буде додано стовпець зі значеннями `5.100`

Time (s)	with(52,t)	attimemax(0,10,a	attimemax(0,10,at1(0,u),t, 'nointerp')
0.0000	5.1000	5.1117	5.1000
0.10000	5.1000	5.1117	5.1000
0.20000	5.1000	5.1117	5.1000
0.30000	5.1000	5.1117	5.1000
0.40000	5.1000	5.1117	5.1000

Побудувати поверхню мембрани в момент t_0 максимального зміщення центру можна за допомогою оператора `at(t0,u)`, для якого значення t_0 слід взяти з оператора `attimemax(0,10,at1(0,u),t)`. Наступна суперпозиція оператрів `at(attimemax(0,10,at1(0,u),t),u)` не працює. Форму мембрани з максимальним зміщенням центру можна було б побудувати по формулі

`at(5.1117,u)`, використовуючи в першому аргументі знайдене вище значення часу t_0. Але доцільніше використати вираз `attimemax(0,10,at1(0,u),u)`. Тоді буде знайдено час t_0, в який досягається максимальне зміщення точки `r=0`, і в цей момент буде обчислюватися змінна `u` у всіх точках `r`.

`Results` **-> Plot Group ->**2D Plot Group

2D Plot Group 6:

Data – **Data set** = `Revolution 1D 1`

`2D Plot Group 6->` Surface

2D Plot Group 6 – Surface 1:

Expression – **Expression** = `attimemax(0,10,at1(0,u),u)`

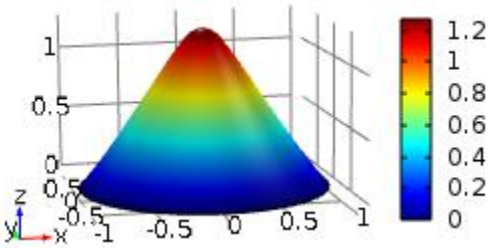
RMB> Height Expression

2D Plot Group 6 – Surface 1 – Height Expression 1:

Axis – **Scale factor** = `True, 1`

LMB>Plot

Буде побудований наступний графік, який практично співпадає побудованим вище на 52-му часовому кроці.



В подальших вправах цього прикладу буде зручно використовувати одновимірні графіки, які будемо будувати на діаметральному (або радіальному) перерізі мембрани. Для них потрібне одновимірне джерело даних, але воно (`Cut Line 2D 1`) вже існує в нашій моделі.

Побудуємо графік розв'язку $u(r,t)$ при `t=3` вздовж діаметрального перерізу.

`Results ->` 1D Plot Group

1D Plot Group 7 :

Data – **Data set** = `Cut Line 2D 1`

`1D Plot Group 7 ->` Line Graph

1D Plot Group 7 - Line Graph 1 :

y-Axis Data – **Expression** = `at(3,u)`

LMB > Plot (верхня крива на наступному рисунку)

Побудуємо графік похідної $\frac{\partial u}{\partial r}(r,t)$ розв'язку при `t=3` вздовж діаметрального перерізу.

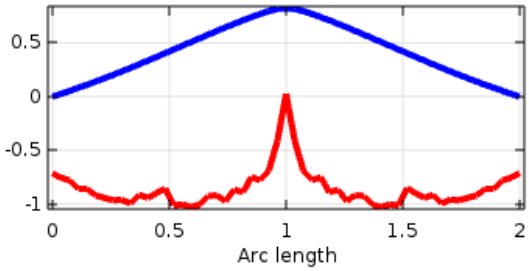
`1D Plot Group 7 ->` Line Graph

1D Plot Group 7 - Line Graph 2 :

y-Axis Data – **Expression** = `d(at(3,u),r)`

(той же графік буде побудований виразом `d(with(31,u),r)`)

LMB > Plot (нижня крива на наступному рисунку)



В системі існують оператори `timemax(t1,t2,expr)` і `timemin(t1,t2,expr)`, які знаходять максимум і мінімум виразу `expr` на вказаному діапазоні часу `t1≤t≤t2`. Якщо вираз `expr` залежить крім змінної часу `t` ще від просторових змінних, то максимум і мінімум знаходиться в кожній точці окремо і повертається просторовозалежний вираз. Наприклад вираз (в умовних позначеннях) `v(r)=timemax(0,10,u(r,t))` повертає максимуми по часу (`0≤t≤10`) в кожній точці `r` розрахункового інтервалу `0≤r≤1`, тобто функцію `r`. Вираз `v(r)` може не співпадати з профілем радіального перерізу мембрани ні в який момент часу. Пояснимо це.

На відрізку `0≤r≤1` побудуємо графіки виразів `at(5.1117,u)` і `timemax(0,10,u)` . Перший з них зображає радіальний переріз мебрани в момент часу `t=5.1117`, коли центральна точка досягає максимального зміщення, а другий – графік максимальних зміщень точок перерізу.

`Results` -> 1D Plot Group

1D Plot Group 8 :

`Data` – **Data set** = `Cut Line 2D 1`

`1D Plot Group 8` -> Line Graph

1D Plot Group 8 - Line Graph 1 :

`y-Axis Data` – **Expression** = `at(5.1117,u)`

`x-Axis Data` – **Parameter, Expression** = `Expression, r`

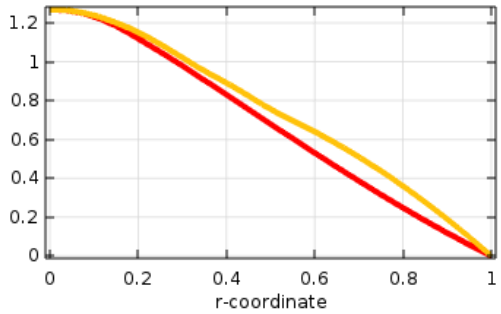
RMB > Duplicate

1D Plot Group 8 - Line Graph 2 :

`y-Axis Data` – **Expression** = `timemax(0,10,u)`

(той же графік буде побудований виразом `timemax(0,10,at1(r,u))`)

LMB > Plot



Як бачите криві не співпадають. Пояснимо це, побудувавши максимальне зміщення деяких точок мембрани. Виконайте наступні інструкції.

Line Graph 2 : RMB > Duplicate

Line Graph 3 :

`y-Axis Data` – **Expression** = `timemax(0,10,at1(0.2,u))`

Цей вираз обчислює максимальне зміщення `u` в точці `x=0.2`, а наступна інструкція зобразить його у вигляді точки.

`x-Axis Data` – **Expression** = `0.2`

`Coloring and Style` – **Width** = `8`

Line Graph 3 : RMB > Duplicate

Line Graph 4 :

`y-Axis Data` – **Expression** = `timemax(0,10,at1(0.4,u))`

`x-Axis Data` – **Expression** = `0.4`

Line Graph 4 : RMB > Duplicate

Line Graph 5 :

`y-Axis Data` – **Expression** = `timemax(0,10,at1(0.6,u))`

`x-Axis Data` – **Expression** = `0.6`

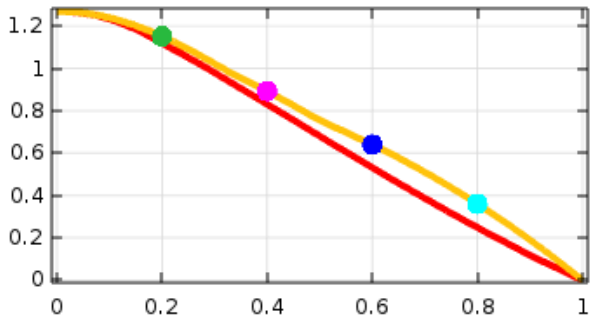
Line Graph 5 : RMB > Duplicate

Line Graph 6 :

`y-Axis Data` – **Expression** = `timemax(0,10,at1(0.8,u))`

`x-Axis Data` – **Expression** = `0.8`

LMB > Plot



Очевидно, що у кожної точки радіального перерізу є максимальні зміщення, але моменти часу, в які вони доягаються, не співпадають. Отже максимальне зміщення точок `r=0.2,0.4,0.6,0.8` досягається в різні моменти часу. Крива виразу `timemax(0,10,u)` проходить через вказані точки і зображає графік максимальних зміщень точок радіального перерізу мембрани, і не зобов'язана співпадати з графіком перерізу в якийсь фіксований момент часу.

Продовжимо ілюструвати роботу операторів.

Оператор `at1(0,u)` обчислює розв'язок в просторовій точці `r=0` і повертає зміщення цієї точки як функцію часу, тобто обчислює $u(0,t)$.

`Results` -> 1D Plot Group

1D Plot Group 9 :

`Data` – **Data set** = `Study 1/Solution 1`

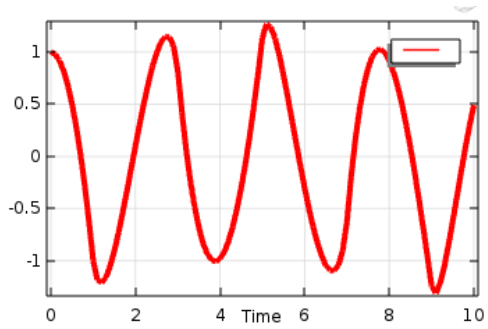
`1D Plot Group 9` -> Global

1D Plot Group 9 - Global 1 :

`y-Axis Data` – **Expression** = `at1(0,u)` (введення в таблиці)

`x-Axis Data` – **Parameter** = `Time`

LMB > Plot



Вираз `timemin(0,10,at1(0,u),'nointerp')` обчислює мінимальне за весь час зміщення (максимальне вниз) центру мембрани (не виконуючи інтерполяцію по часу).

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 3:

`Expression` – **Expression** = `timemin(0,10,at1(0,u),'nointerp')`

LMB > Evaluate – New Table

Створюється таблиця `Table 2` з двома стовпцями: моментами часу і єдиним значенням `-1.3109` мінімального зміщення центру.

Time (s)	timemin(0,10,at1(0,u),'nointerp')
0.0000	-1.3109
0.10000	-1.3109
0.20000	-1.3109
0.30000	-1.3109
0.40000	-1.3109

Вираз `timeint(0,10,at1(0,u))` обчислює інтеграл в точці `r=0`, тобто $\int_0^{10} u(0,t)\,dt$.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 4:

`Expression` – **Expression** = `timeint(0,10,at1(0,u))`

LMB > Evaluate – Table 2

В таблицю `Table 2` буде додано стовпець зі значеннями `-0.38995` вказаного інтеграла.

Time (s)	timemin(0,10,at1(0	timeint(0,10,at1(0,u))
0.0000	-1.3109	-0.38995
0.10000	-1.3109	-0.38995
0.20000	-1.3109	-0.38995
0.30000	-1.3109	-0.38995
0.40000	-1.3109	-0.38995

Вираз `timeavg(0,10,at1(0,u))` обчислює середнє за часом значення зміщення в точці `r=0`, тобто $\frac{1}{10}\int_0^{10} u(0,t)\,dt$.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 5:

`Expression` – **Expression** = `timeavg(0,10,at1(0,u))`

LMB > Evaluate – Table 2

Time (s)	timemin(0,10,at1(	timeint(0,10,at1(0,u	timeavg(0,10,at1(0,u))
0.0000	-1.3109	-0.38995	-0.038995
0.10000	-1.3109	-0.38995	-0.038995
0.20000	-1.3109	-0.38995	-0.038995
0.30000	-1.3109	-0.38995	-0.038995
0.40000	-1.3109	-0.38995	-0.038995

Вираз `at(3,at1(0.5,u))` обчислює зміщення точки `r=0.5` в момент часу `t=3`.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 6:

`Expression` – **Expression** = `at(3,at1(0.5,u))`

LMB > Evaluate – Table 2

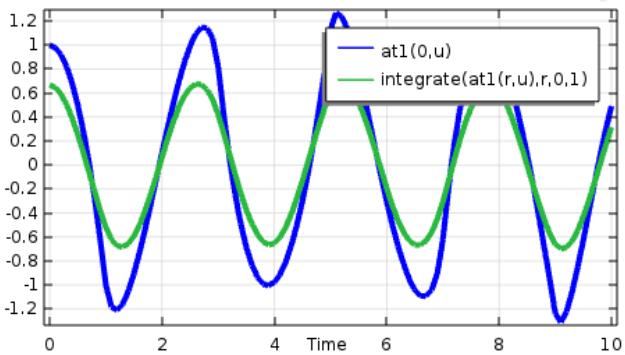
Time (s)	timemin(0,10,	timeint(0,10,at	timeavg(0,10,at1	at(3,at1(0.5,u))
0.0000	-1.3109	-0.38995	-0.038995	0.41929
0.10000	-1.3109	-0.38995	-0.038995	0.41929
0.20000	-1.3109	-0.38995	-0.038995	0.41929
0.30000	-1.3109	-0.38995	-0.038995	0.41929
0.40000	-1.3109	-0.38995	-0.038995	0.41929

Вираз `integrate(at1(r,u),r,0,1)` обчислює інтеграл $\int_0^1 u(r,t)\,dr$, який є функцією часу. Побудуйте його графік, додавши в таблицю виразів вузла **1D Plot Group 9 - Global 1** новий рядок :

`y-Axis Data` – **Expression** = `integrate(at1(r,u),r,0,1)`

LMB > Plot

Самостійно налаштуйте легенду графіка.



Вираз `timeavg(0,0.5,integrate(at1(r,u),r,0,1))` обчислює середнє за часом `0≤t≤0.5` попереднього інтеграла, тобто $\frac{1}{0.5}\int_0^{0.5}\left(\int_0^1 u(r,t)\,dr\right)dt$.

`Results` -> **`Derived Values`** – Global Evaluation

Derived Values - Global Evaluation 7:

`Expression` – **Expression** =
`timeavg(0,0.5,integrate(at1(r,u),r,0,1))`

LMB > Evaluate – Table 2

Time (s)	timemin(0,10,	timeint(0,10,at	timeavg(0,10,at1	at(3,at1(0.5,u))	integrate(at1(r	timeavg(0,0.5,integr
0.0000	-1.3109	-0.38995	-0.038995	0.41929	0.66667	0.52202
0.10000	-1.3109	-0.38995	-0.038995	0.41929	0.64729	0.52202
0.20000	-1.3109	-0.38995	-0.038995	0.41929	0.59209	0.52202
0.30000	-1.3109	-0.38995	-0.038995	0.41929	0.50536	0.52202
0.40000	-1.3109	-0.38995	-0.038995	0.41929	0.39181	0.52202

Майте на увазі, що останнє обчислення подвійного інтеграла триває досить довго, і збільшення часового інтервалу `0≤t≤0.5` впливає на це суттєво.

Якщо вираз expr в операторі v=timeavg(t1,t2,expr) залежить від просторових координат, наприклад, expr=u(r,t), то середнє за часом обчислюється в кожній точці r розрахункового інтервалу, і вираз v буде функцією r. Перевірка цього виконується аналогічно до того, що ми робили з оператором timemax(0,10,u) – будували криву і окремі точки. Але тут для побудови точкового графіка ми використаємо точкове джерело даних.

Results -> **Data Set- > More Data Sets ->** Cut Point 1D

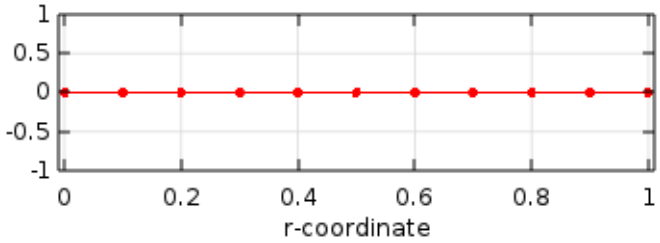
Data Sets - Cut Point 1D 1:

Data – **Data set** = Study 1/Solution 1

Point Data –**Entry method** = Coordanates

Point Data – **r** = range(0,0.1,1)

LMB > Plot



Спочатку побудуємо криву середніх зміщень точок радіального перерізу за інтервал часу 0≤t≤3.

Results -> 1D Plot Group

1D Plot Group 10 :

Data – **Data set** = Cut Point 1D 1

1D Plot Group 10 -> Line Graph

1D Plot Group 10 - Line Graph 1 :

Data – **Data set** = Cut Line 2D 1

y-Axis Data – **Expression** = timeavg(0,3,u)

x-Axis Data – **Parameter, Expression** = Expression, r

Coloring and Style – **Color** = Red

Потім побудуємо середні зміщення точок, які належать точковому ДД, за той же інтервал часу 0≤t≤3.

1D Plot Group 10 -> Point Graph

1D Plot Group 10 - Point Graph 1 :

Data – **Data set** = From parent

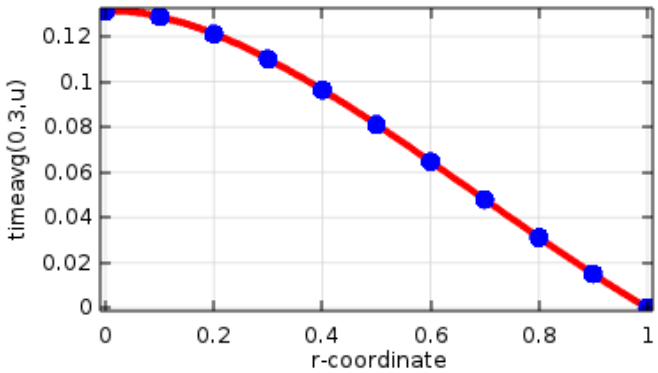
y-Axis Data – **Expression** = timeavg(0,3,u)

Тут вираз timeavg(0,3,u) обчислює середнє за часом 0≤t≤3 зміщення точок, визначених в точковому джерелі даних.

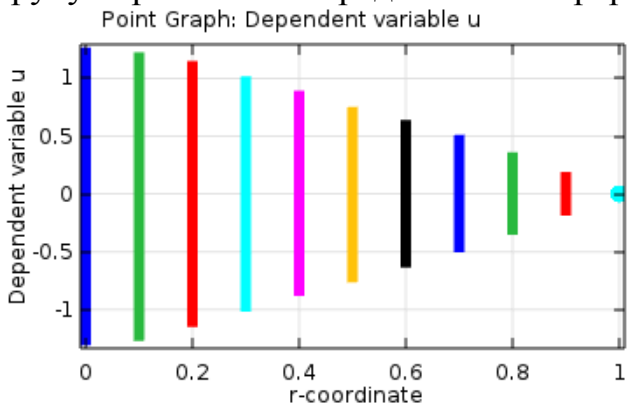
x-Axis Data – **Parameter, Expression** = Expression, r

Coloring and Style – **Color, Width** = Blue, 7 (*два поля*)

LMB > Plot



Є ще одне застосування точкового джерела даних. На наступному рисунку зображені траєкторії руху окремих точок радіального перерізу.



Покажемо, як такий графік було побудовано.

`Results` –> 1D Plot Group

1D Plot Group 11 :

`Data` – **Data set** = `Cut Point 1D 1`

`Data` – **Time selection** = `All`

`1D Plot Group 11` –> Point Graph

1D Plot Group 11 - Point Graph 1 :

`Data` – **Data set** = `From parent`

`y-Axis Data` – **Expression** = `u`

`x-Axis Data` – **Parameter, Expression** = `Expression, r`

`Coloring and Style` – **Width** = `6`

LMB > Plot (*попередній рисунок*)

Якщо у вузлі **1D Plot Group 11** в полі **`Data`** – **Times (s)** ввести конкретний момент часу `t1`, то ви побудуєте положення точок в обраний момент. Змініть, наприклад, налаштування у ПВ вузла **1D Plot Group 11:**

`Data` – **Time selection** = `Manual`

`Data` – **Times (s)(1-101)** = `5`

LMB > Plot

Ви отримаєте графік, який показано на наступному рисунку ліворуч.

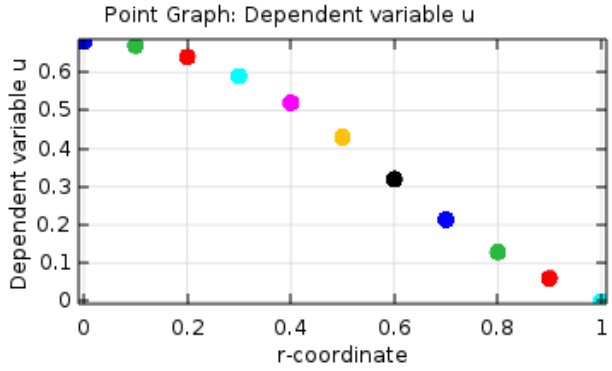
Якщо, в полі **Times** задати кілька моментів часу, то будуть побудовані вертикальні відрізки, які зображають траєкторії точок між першим і останнім введеними моментами. Наступний рисунок праворуч зображає графік, який отримано при налаштуваннях у ПВ вузла **1D Plot Group 11**, які виділяють

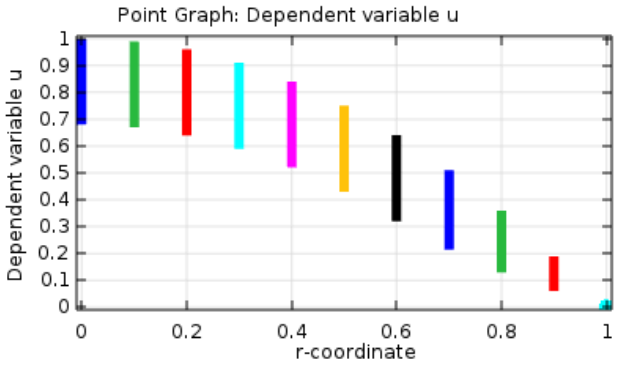
інтервал часу від першого до п’ятого моментів (з послідовності `range(0,0.1,10)`).

`Data` – **Time selection** = `Manual`

`Data` – **Times (s)(1-101)** = `1,5`

LMB > Plot





Суттєвим моментом при побудові останніх чотирьох графіків є значення поля **`x-Axis Data`** – **Expression** =`r`.

■

Важливими елементами COMSOL Multiphysics є списки значень. Поля, біля яких стоїть кнопка , вимагають введення списків. Їх можна вводити по-різному:

- використовуючи кнопку списку ;
- використовуючи введення чисел через проміжок або через кому, наприклад, `10, 15, 23, 29.7, 30;`
- використовуючи числа з проміжками-розділювачами всередині фігурних дужок, наприклад, `{0 0.25 0.5 0.75 1};`
- використовуючи список в фігурних дужках в якості аргументу функції, наприклад, `cos({0 pi/6 pi/4 pi/3 pi/2});`
- для введення рівномірно розподілених значень використовують функцію `range(start,step,stop)`, де аргументи `start` і `stop` представляють початкове і кінцеве значення елементів списку, а `step` – інтервал між елементами.

Зазначені способи можна поєднувати. Наведемо кілька прикладів.

1. Запис

   `range(0,0.2,1) 1.5  2.0.`

   обробляється як список `0 0.2 0.4 0.6 0.8 1.0 1.5 2.0.`
2. Запис (дві функції `range` через проміжок)

   `range(0,0.2,1.0) range(2,0.1,2.5)`

   обробляється як об'єднаний список

   `0  0.2  0.4  0.6  0.8  1.0  2.0  2.1  2.2  2.3  2.4  2.5.`
3. Запис `2*range(0,0.2,1.0)`

   призводить до створення списку `0  0.4  0.8  1.2  1.6  2.0.`
4. Запис `2^range(0,1,5)` призводить до створення списку `1 2 4 8 16 32.`

В системі COMSOL Multiphysics важливим елементом синтаксису виразів і функцій є одиниці вимірювання фізичних величин. Більшість функцій користувача і вбудованих математичних функцій використовують безрозмірні аргументи і повертають безрозмірні значення. Тому виклик функцій з аргументом, який має розмірність, в ряді контекстів буде приводити до помилки. Наприклад, це може статися в виразі `sin(t)`, де `t` – змінна часу. Справа в тому, що змінна `t` зазвичай має розмірність секунди, якщо в моделі використовується система одиниць СІ (SI). Щоб використовувати змінну часу `t` в якості безрозмірного аргументу її слід доповнити розмірністю `[1/s]`. Тоді розмірність `[s]` (секунда) змінної `t` множиться на розмірність `[1/s]` і величина `t[1/s]` стає безрозмірною. В результаті, виклик функції `sin(t[1/s])` буде коректним. При некоректному використанні розмірності COMSOL виділяє вираз помаранчевим кольором.

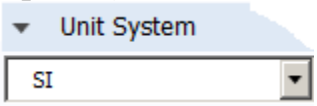
Множення розмірностей має пріоритет над математичними операціями. Наприклад, вираз `1/2[m]` обчислюється як `1/(2[m])` і результатом буде величина `0.5[1/m] = 0.5[m`$^{-1}$`]`. Тоді як `(1/2)[m]` і `1/2*1[m]` представляють величину `0.5[m]=50[sm]`. Аналогічно, якщо `L=2[m]`, то величина `L[1/s]` представляє швидкість `2[m/s]`.

Множення розмірностей можна використовувати при додаванні для отримання сумісних по типу доданків. Наприклад, якщо змінна `rho` представляє густину, то допустимим є вираз `10[kg]+rho[m^3]`. Крім того, множення використовується для з'єднання декількох одиниць, наприклад, значення `4.5[N][m]` еквівалентно `4.5[N*m]=4.5[J]` (`[J]` – джоуль).

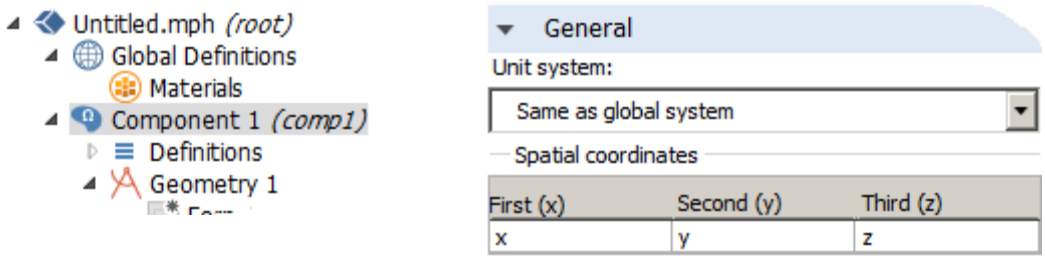
При додаванні двох величин однакової розмірності (по типу), наприклад довжини, але заданих в різних одиницях, система COMSOL перетворює результат в основну систему одиниць. Наприклад, вираз `0.5[m]+25[sm]` перетвориться до значення `0.75[m]`, якщо СІ (SI) є основною системою одиниць моделі.

До будь-якої величини можна додавати безрозмірну константу. Тоді система вважає, що вона (константа) має такий же тип розмірності, як перший операнд, і при цьому її розмірність відповідає основній системі одиниць моделі.

Сама система одиниць моделі визначається в ПВ кореневого вузла. На початку роботи з файлом цей вузол має ім’я **Untitled.mph (root)**. В його ПВ в списку **`Unit system`** можна обрати (або відключити) бажану систему одиниць.



Кожний компонент моделі теж має власну систему одиниць, яка визначається в ПВ вузла **Component** в списку **Unit system** розділа **`General`**. Найчастіше нас задовольняє вибір `Same as global system`, тобто система одиниць компонента така сама, яка обрана для всієї моделі. Тут же задаються імена змінних для просторових координат.



1.8. Довідкова система і документація

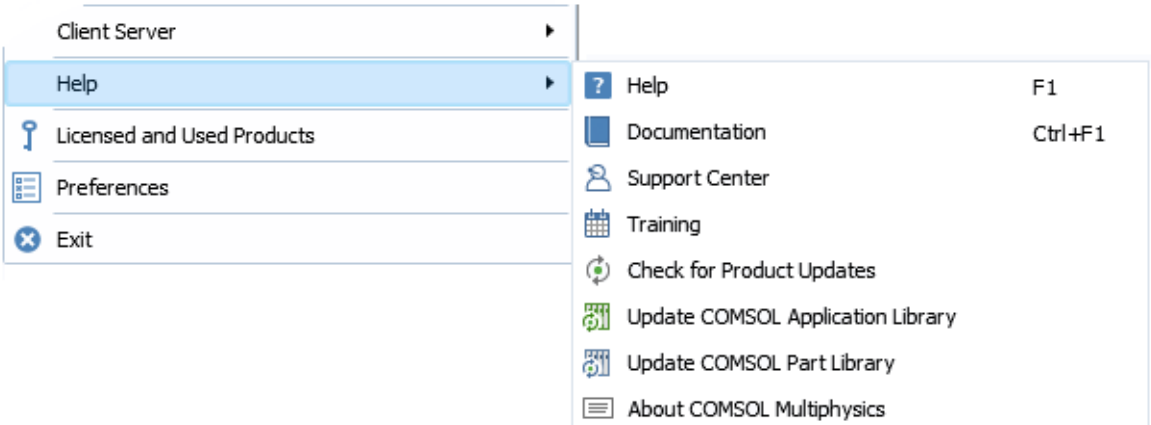
Система COMSOL Multiphysics постачається з великою кількістю електронної документації. Вона встановлюється разом з COMSOL в двох форматах - PDF і HTML. Її ви можете знайти у встановлювальній папці системи COMSOL в каталозі `..\doc`.

Довідкова система відкривається при натисканні мишею на кнопку Help, що розташована в правому верхньому куті вікна програми COMSOL Multiphysics, або натисканням на клавішу `F1` клавіатури.

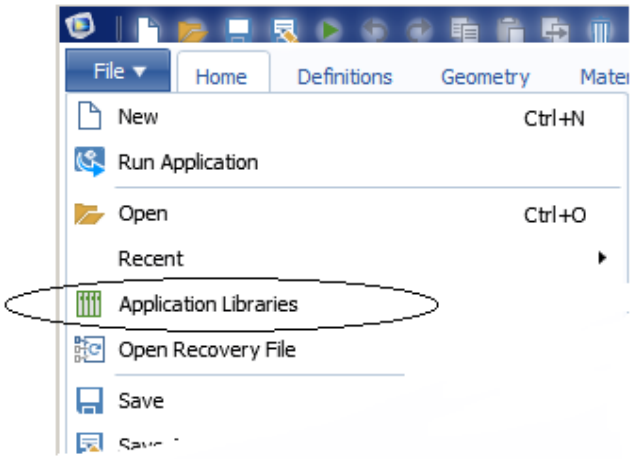
В результаті в правій частині вікна програми відкривається панель довідки з темою, пов'язаною з активним вузлом дерева моделі.

Комбінація клавіш `Ctrl-F1` відкриває окреме довідкове вікно. В ньому міститься список всієї документації COMSOL Multiphysics в форматах HTML і PDF. Це вікно працює як самостійний Browser.

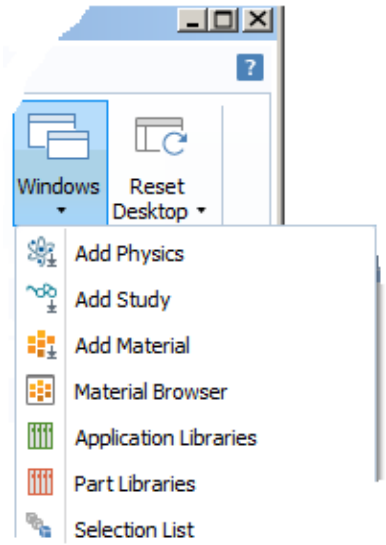
Крім того, в меню `File` є пункт `Help`, який містить ті ж команди роботи з контекстною довідкою (`F1`) і з документацією (`Ctrl-F1`).



Також в меню `File` є пункт Application Libraries, який відкриває бібліотеку готових моделей/прикладів з різних областей фізики і техніки.



Ця команда є також на закладці Home стрічки інструментальних панелей в випадному списку Windows.



За цією командою відкривається панель Application Libraries. Вибір в списку приклада приводить до завантаження mph файлу, який ви можете аналізувати, змінювати і виконувати.

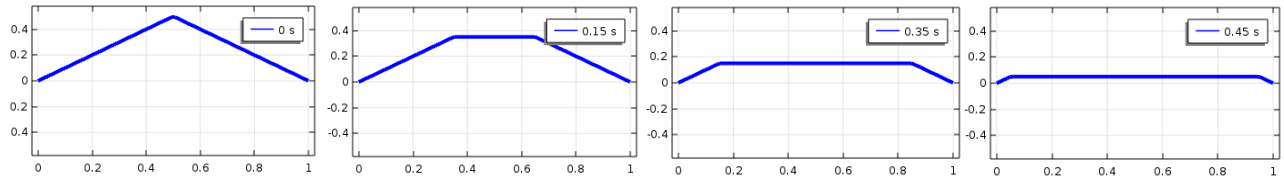
Команда Part Libraries відкриває бібліотеку деталей, тобто готових геометричних об'єктів, які при завантаженні в гілку **Geometry** створюють відповідні вузли. Ви можете не креслити свої 3D об'єкти, перерізи, канали і т.д., а використовувати готову геометрію.

На сайті розробників https://www.comsol.com/models є галерея моделей. Вона містить документацію (зазвичай в форматі PDF) по великій кількості прикладів.

1.9. Задачі для самостійного розв’язання.

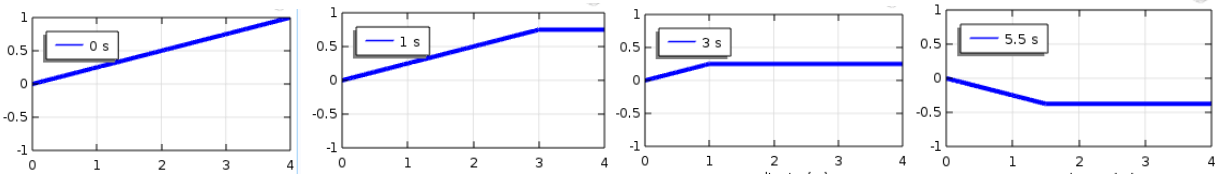
Задача 1. Дослідити коливання скінченної струни довжини L=1 з закріпленими кінцями, початковим зміщенням $u(x,0)=0.5-|x-0.5|$, і нульовою початковою швидкістю $u'_t(x,0)=0$. Покладіть a=1 (швидкість поширення хвиль).

Побудуйте модель, в якій створіть графіки форми струни в різні моменти часу. На наступному рисунку показана струна в початковий і кілька наступних моментів часу.



Задача 2. Дослідити коливання скінченної струни довжини L з одним закріпленим кінцем і іншим вільним. В такій постановці граничні умови мають вигляд $u(0,t)=0, u'_x(L,t)=0$. Покладіть a=1 (швидкість поширення хвиль), L=4, $u(x,0)=x/4$ (початкове зміщення), $u'_t(x,0)=0$ (початкова швидкість).

На наступному рисунку наведені графіки розв’язку `u(x,t)` в моменти часу `t=0,1,3,5.5`.



Задача 3. Дослідити вільні осесиметричні коливання кругової мембрани радіусу R=1 (див. параграф 1.6.3), яка закріплена на контурі, має початкову форму $u(r,0)=1-r$ $(0\le r\le 1)$ і нульову початкову швидкість $u'_t(r,0)=0$. В задачі покладіть $a=1$ (швидкість поширення хвиль). Побудуйте анімацію руху лінії перерізу поверхні мембрани вертикальною площиною.

Задача 4. Обчислити і зобразити стаціонарний прогин балки.

Диференціальне рівняння осі зігнутої балки, що знаходиться під дією зовнішніх сил, має вигляд $EJ\,u''''_{xxxx}=q(x)$, де q – зовнішня сила (розрахована на одиницю довжини), E і J – модуль пружності матеріалу балки і момент інерції її перерізу. Ось X збігається з віссю балки. Функція `u(x)` являє вертикальне зміщення точок осі балки, що мали до навантаження горизонтальну координату `x`. Для вільно опертої на краях балки граничні умови мають вигляд `u[0]=0,u[L]=0, u''[0]=0, u''[L]=0`, де `L` – довжина балки. В задачі покладіть `L=1, E=1, J=1, q(x)=-1`.

Задача 5. Дослідити процес поширення тепла в тонкому стрижні бокова поверхня якого теплоізольована.

Задача поширення тепла в тонкому однорідному стрижні, бокова поверхня якого теплоізольована (тепло може поширюватися тільки вздовж його осі),

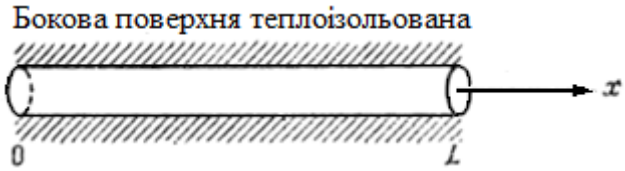
математично формулюється так: знайти обмежену функцію $u(x,t)$ $(t>0,\, 0\le x\le L)$, що задовольняє рівнянню теплопровідності

$$\frac{\partial u}{\partial t}=a^2\frac{\partial^2 u}{\partial x^2}\quad (t>0,\, 0<x<L),$$

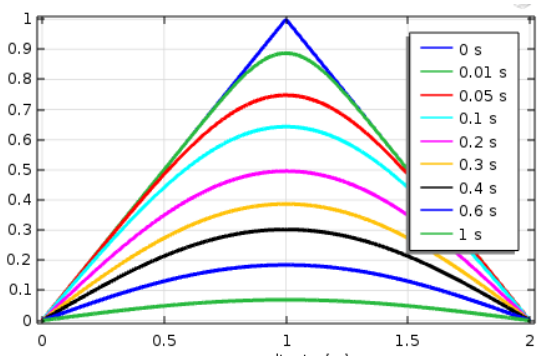
початковій умові

$$u(x,0)=\varphi(x)\quad (0<x<L),$$

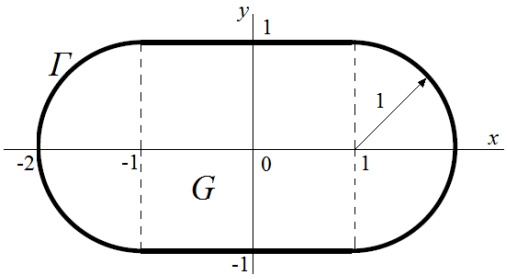
і граничним умовам $u(0,t)=u_0$, $u(L,t)=u_1$, де $u(x,t)$ температура перерізу x стрижня в момент часу t (стрижень тонкий, це означає, що температура точок будь-якого поперечного перерізу x однакова).



Виконати дослідження при L=2, a=1, нульовій граничній температурі $(u(0,t)=0,\, u(L,t)=0)$, і початковій температурі $u(x,0)=\varphi(x)=1-|1-x|$.



Задача 6. Розв'яжіть рівняння Пуассона $u''_{xx}+u''_{yy}=-1$ в області овалу G, який показано на наступному рисунку. Функція $u(x,y)$ задовольняє нульовим граничним умовам Дірихле $u_\Gamma=0$ на його контурі Γ .



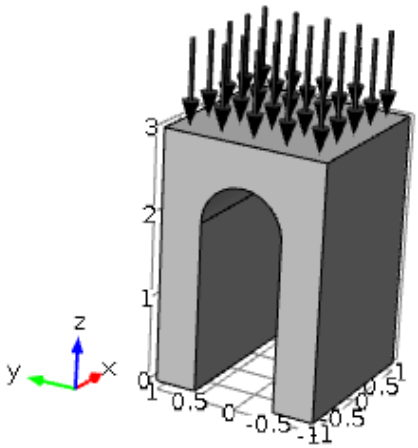
Задача 7. Розв'яжіть рівняння Пуассона (9) $u''_{xx}+u''_{yy}=-4$ в одиничному колі G з однорідними граничними умовам $u_{\partial G}=0$. Побудуйте графік функції $u(x,y)$.

Точним розв'язком є функція $u(x,y)=1-x^2-y^2$. Знайдіть абсолютну похибку чисельного розв'язку, та побудуйте графік різниці двох розв'язків.

Задача 8. Дослідити процес охолодження квадратної пластини $[0,1]\times[0,1]$, контур якої підтримується при нульовій температурі, а початкова температура дорівнює одиниці $(u(x,y,0)=1)$. Виконати дослідження при одиничному значенні температуропровідності.

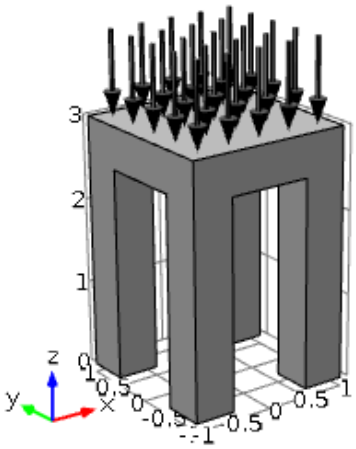
Побудувати графіки: залежності температури від часу в центрі пластини; залежності середньої температури пластини від часу; залежності середньої температури на діагоналі пластини від часу. Побудувати зонди, які відслідковують середню температуру пластини і температуру в фіксованій точці.

Задача 9. Розв'язати задачу деформування тривимірного тіла, яке показано на наступному рисунку. Воно являє бетонний блок $-1 \le x, y, \le 1,\ \ 0 \le z \le 3$ з отвором у вигляді арки, яка розташована вздовж осі X. Радіус арки 0.5 м. Інші параметри геометрії візьміть з рисунка. Основа тіла закріплена. Зверху діє сила $F = 20000\, н/м^2$, спрямована вертикально вниз і рівномірно розподілена по верхній грані.



Побудувати графіки: деформований поверхневий графік напруження σ_z; поверхневий графік напруження σ_z зі схованими фронтальними гранями; зобразити напруження σ_z по двох горизонтальних перерізах, розташувавши їх графіки поруч; зобразити картину навантаження; побудувати деформований лінійний графік величини переміщення на ребрах тіла; показати точки максимуму і мінімуму σ_z на поверхні арки.

Задача 10. Розв'язати задачу деформування тривимірного тіла, яке показано на наступному рисунку. Воно являє бетонний блок $-1 \le x, y, \le 1,\ \ 0 \le z \le 3$ з прямокутними отворами завширшки 1 м і заввишки 2.5 м, які розташовані вздовж координатних осей X та Y. Інші параметри геометрії візьміть з рисунка. Основа тіла закріплена. Зверху діє сила $F = 20000\, н/м^2$, спрямована вертикально вниз, і яка рівномірно розподілена по верхній грані.



Побудувати графіки: деформований поверхневий графік напруження σ_z; поверхневий графік напруження σ_z зі схованими фронтальними гранями; зобразити напруження σ_z по двох горизонтальних перерізах, розташувавши їх графіки поруч; зобразити картину навантаження; побудувати деформований лінійний графік величини переміщення на ребрах тіла; показати точки максимуму і мінімуму σ_z на площині $z = 2.5$.