

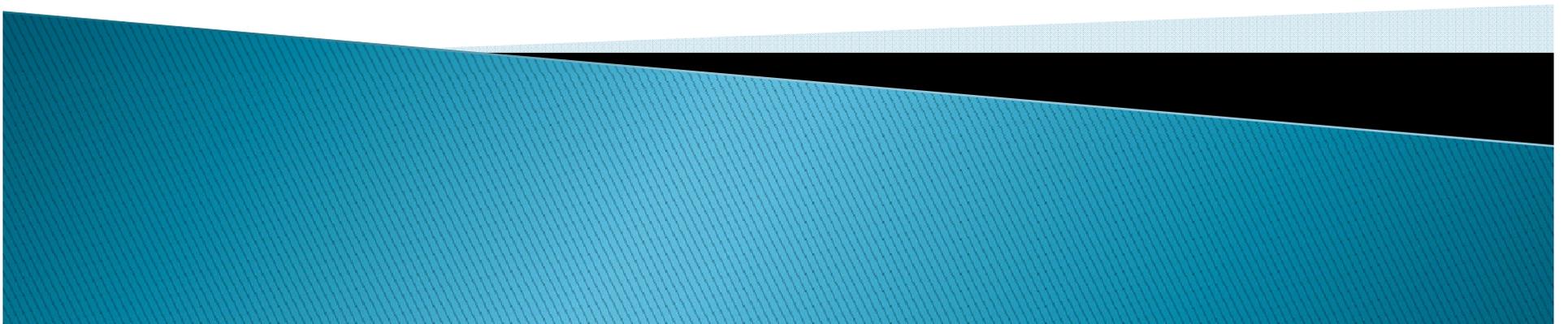


Instituto Nacional de Astrofísica, Óptica y Electrónica

Coordinación de Ciencias Computacionales
Tonantzintla, Puebla

Diseño de Software Científico a Través de Patrones

Gustavo Rodríguez Gómez



Contenido

- ▶ Introducción
- ▶ Problemas con el Software Científico
- ▶ Patrones de Software
- ▶ Ejemplo
- ▶ Preguntas

Contenido

- ▶ *Introducción*
- ▶ Problemas con el Software Científico
- ▶ Patrones de Software
- ▶ Ejemplo
- ▶ Preguntas

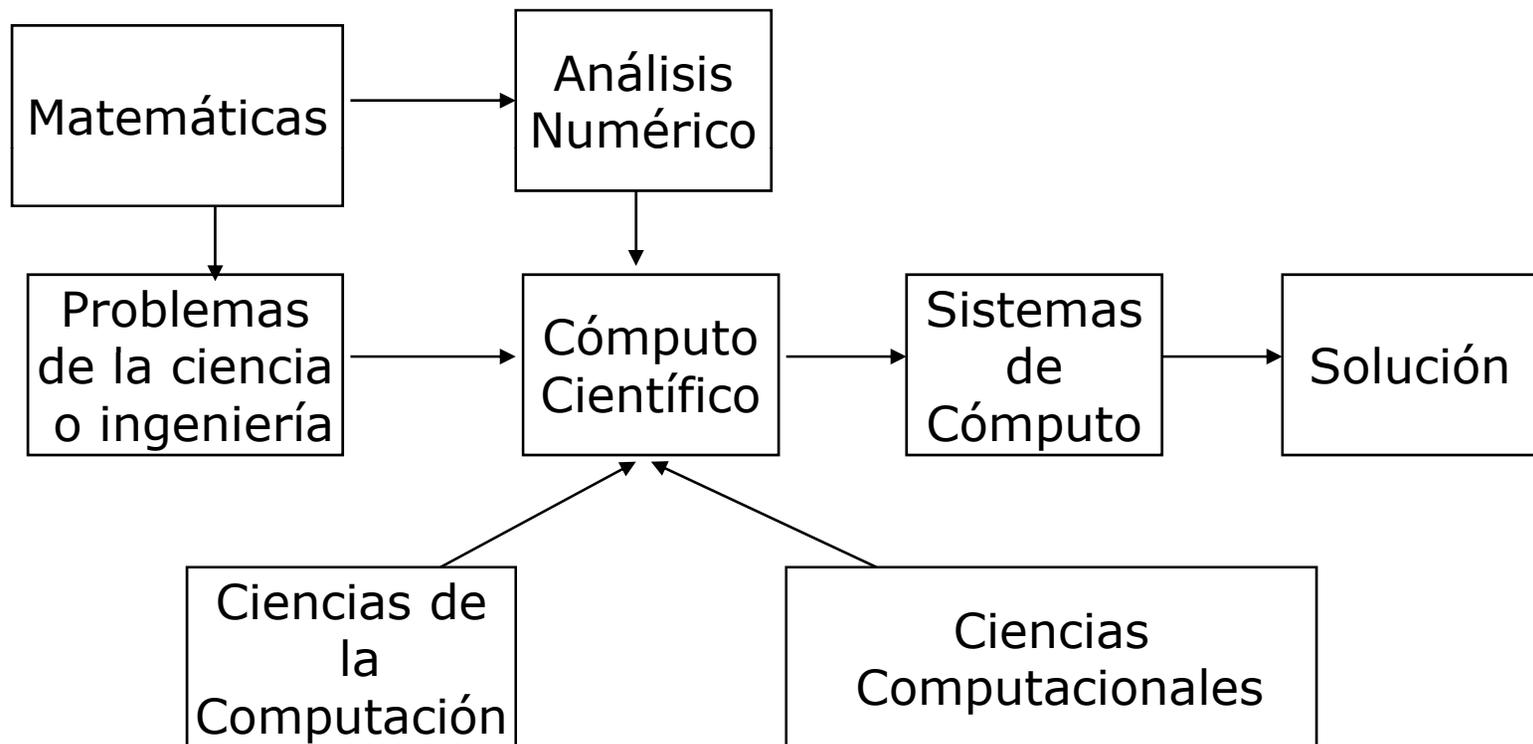
Introducción

- ▶ Actualmente no hay área de la ciencia o de la ingeniería que no emplee computadoras para simular modelos.
- ▶ Las técnicas usadas para resolver problemas científicos y tecnológicos son parte de la Ciencia Computacional (CC).
- ▶ El empleo de las técnicas que nos permiten profundizar en los problemas científicos y tecnológicos es llamada *Ciencia Computacional*.

Computación Científica

- ▶ La computación científica es la colección de herramientas, técnicas y teorías requeridas para resolver en una computadora modelos matemáticos de problemas de la ciencia e ingeniería.

Áreas Relacionadas a la Computación Científica



Computación Científica

- ▶ La *computación científica* es inherentemente multidisciplinaria, involucra:
 - Formulación de un modelo matemático.
 - El análisis matemático del modelo.
 - Reducción del modelo a una forma apropiada para la simulación numérica.
 - Codificación correcta del modelo en una computadora.
 - Explotación adecuada de la arquitectura de la computadora.

Computación Científica

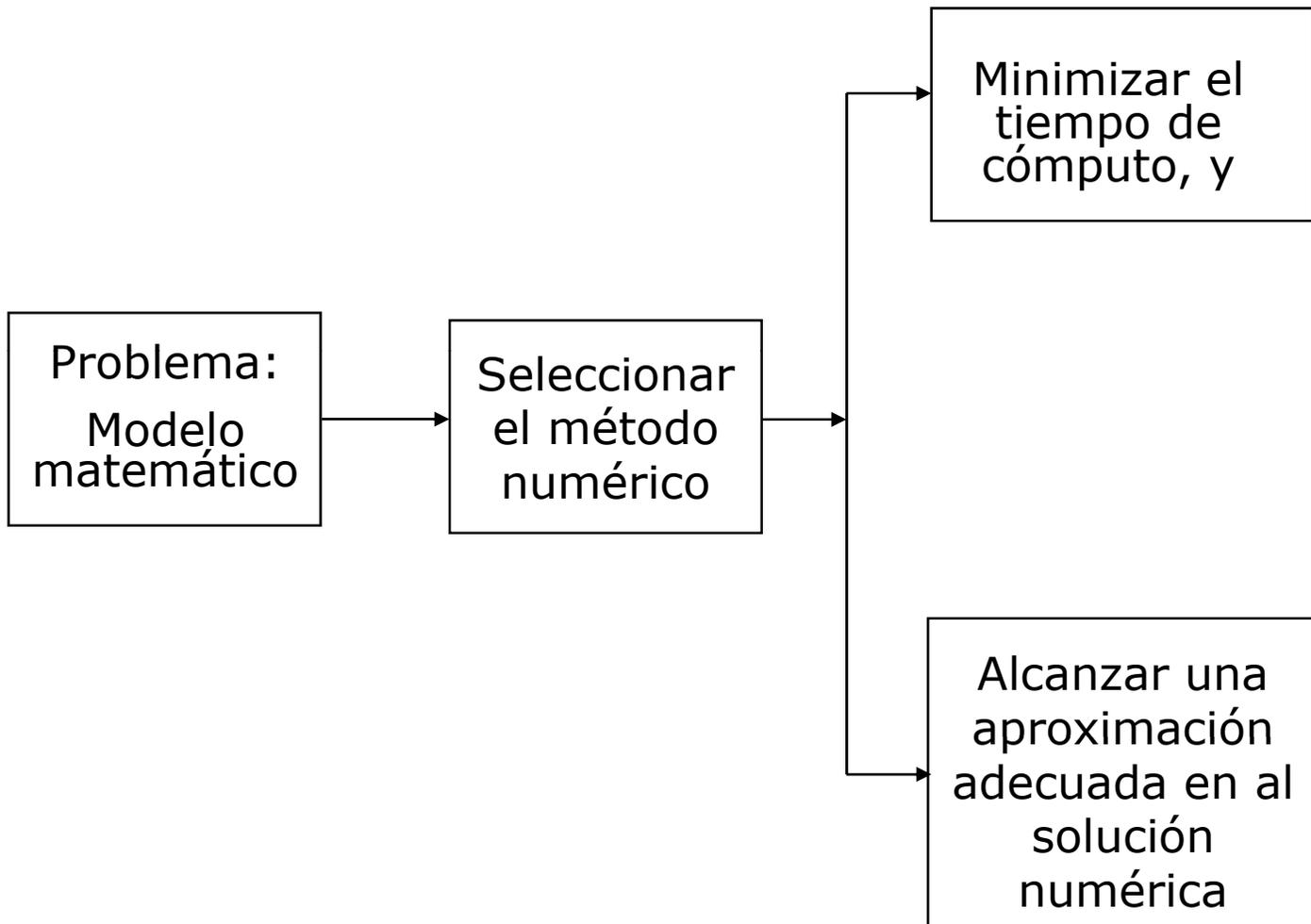
- ▶ La *computación científica* está íntimamente relacionada con la Ciencias de la Computación:
 - *Algoritmos y Estructura de Datos*: utilización eficiente de los algoritmos, manejo adecuado de la estructura de datos.
 - *Ingeniería de Software*: desarrollo del software científico con los atributos de calidad adecuados.
 - *Métodos de Visualización*: procesamiento de datos científicos y presentación de estos en forma “simple”.

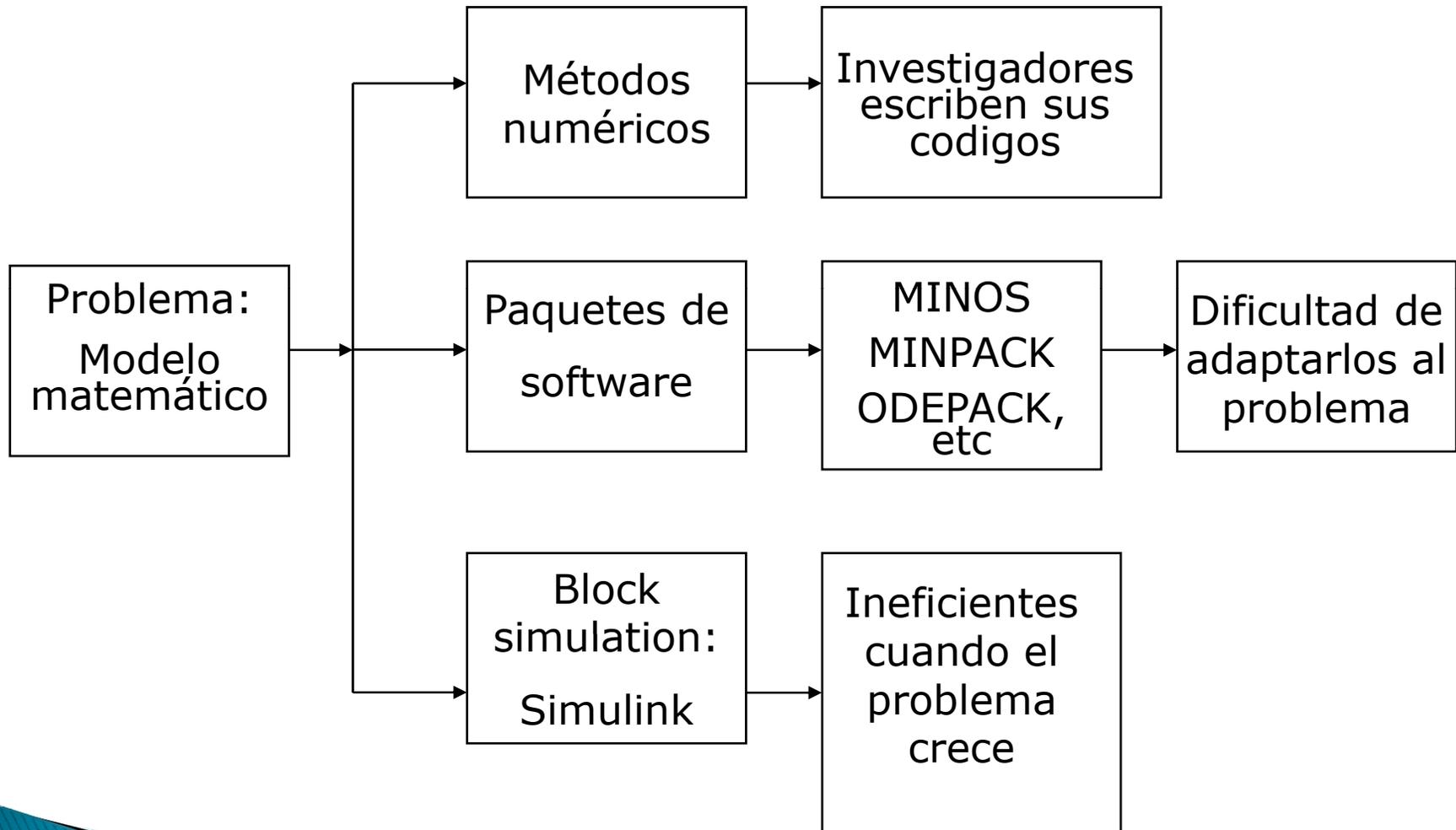
Computación Científica

- ▶ La *computación científica* está íntimamente relacionada con el
 - *Análisis Numérico*: el estudio de los algoritmos para los problemas de las matemáticas continuas.

Contenido

- ▶ Computación Científica
- ✓ *Problemas con el Software Científico*
- ▶ Patrones de Software
- ▶ Ejemplo
- ▶ Preguntas





Problemas con el Software Científico

- ▶ Los métodos numéricos se encuentran en bibliotecas estáticas.
- ▶ No hay flexibilidad para introducir cambios:
 - añadir nuevos métodos,
 - extender las capacidades del algoritmos,
 - cambios en los formatos de llamada.
- ▶ Dificultad de adaptar los problemas de aplicación específicos a los paquetes.

Problemas con el Software Científico

- ▶ Origen de los problemas:
 - Lenguajes por procedimientos (procedural languages).
 - Acoplan los algoritmos numéricos a la representación específica de lo datos.

Solución Propuesta

- ▶ Utilizar el paradigma orientado a objetos aplicando patrones de cómputo científico y patrones de diseño de software para el desarrollo de la programación de las estrategias numéricas.

Contenido

- ▶ Introducción
- ▶ Problemas con el Software Científico
- ▶ *Patrones de Software*
- ▶ Ejemplo
- ▶ Preguntas

Patrones

*Cada patrón describe un problema que existe una y otra vez en nuestro entorno, y describe la esencia de la solución a ese problema, de tal modo que pueda utilizarse esa solución un millón de veces más, sin siquiera hacerlo de la misma manera dos veces.
(Christopher Alexander)*

- ▶ Un PATRÓN es:
 - Una solución a un problema en un contexto particular
 - Son sistemas de objetos que se repiten una y otra vez en la arquitectura de software y su implementación.
 - "Los patrones de diseño ayudan a obtener un plano para conseguir una gran reusabilidad y mantenibilidad en las aplicaciones". [[Nielsen96](#):78]

Scientific Computing Patterns

Contexto práctico

Necesidades comunes

+

Buenas soluciones

Podemos emplear la misma idea en otros contextos

- ▶ Un medio para registrar la experiencia
- ▶ Una forma de compartir conocimiento
- ▶ Un lenguaje común
- ▶ Facilitan el reuso de diagramas conceptuales, diseños y arquitecturas.

Contenido

- ▶ Introducción
- ▶ Problemas con el Software Científico
- ▶ Patrones de Software
- ▶ *Ejemplo*
- ▶ Preguntas

Problema Matemático

- ▶ Se desea aproximar numéricamente la solución de un sistema de ecuaciones diferenciales ordinarias (sedo) con valores iniciales.
- ▶ La solución se aproximará por medio de los algoritmos de la clase Runge–Kutta de R-etapas.

Patrones De Cómputo Científico

- ▶ Estrategia de solución:
 - Identificar patrones de cómputo científico.
 - A partir de estos se diseña la “máquina resolvidora”
 - Utilizar patrones de diseño disponibles en la comunidad de software.

Características del Diseño.

- ▶ Características del diseño
 - La estructura de la clase integradora debe ser reusable sin modificación.
 - La clase integradora no debe contener detalles del modelo integrador.
 - El modelo matemático debe estar afuera de la clase integradora.
 - La clase integradora debe ser extensible.
 - Los métodos numéricos son reusables para diferentes modelos matemáticos.
 - El método numérico es instanciado al tiempo de compilación.

Model-Solver Pattern

Name: Model-Solver

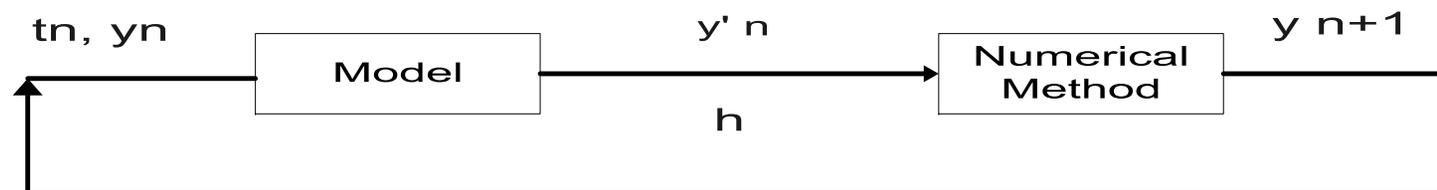
Problem: A physical system represented by a mathematical model, involving systems of algebraic-differential equations, is required to be solved numerically in order to analyze the model behavior.

Intent: Give a feedback between the model and the solver module during the simulation loop.

Context: The implementation of continuous process simulation is well-defined problem by an ordinary differential equations system, which is solved by an iterative fashion during the execution time.

Forces: The decoupling between models and numerical methods increases extensibility. The models are not aware of the numerical issues

Solution: Specify the mathematical model and the solver by modules, and define the communication between them in order to give continuity to the simulation loop.

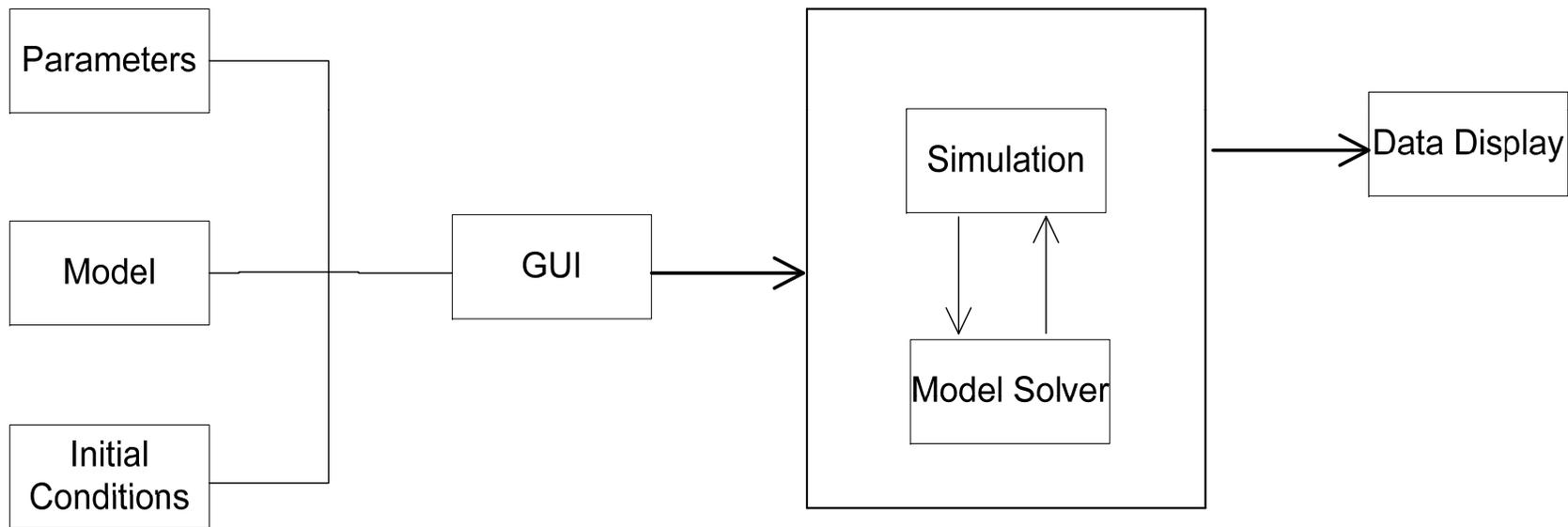


Rationale: The model and the solver separation by modules allow their reusability and their weak coupling.

Descomposición–Sistema

- ▶ Problema
 - Se desea tener un sistema flexible para ejecución de simulaciones.
- ▶ Propósito
 - Proponga un modelo conceptual de simulador de procesos continuos en términos de módulos.
- ▶ Contexto
 - El sistema está representado por ecuaciones algebraicos–diferenciales. Los parámetros de entrada del sistema a simularse están claramente identificados.
- ▶ Fuerzas
 - Es posible descomponer el modelo del sistema a simularse en módulos separables para poder identificar nuevos subsistemas. La solución es “ocultada” al usuario lo cual da gran flexibilidad para seleccionar nuevas estrategias de solución.
- ▶ Solución
 - El modelo de simulación debe permitir interacciones entre las entradas y salidas, modelo y modulo de solución para obtener la información necesaria durante la ejecución de la simulación.
- ▶ Justificación
 - El modelo del proceso continuo a simularse es posible especificarlo en términos de un conjunto de módulos independientes .

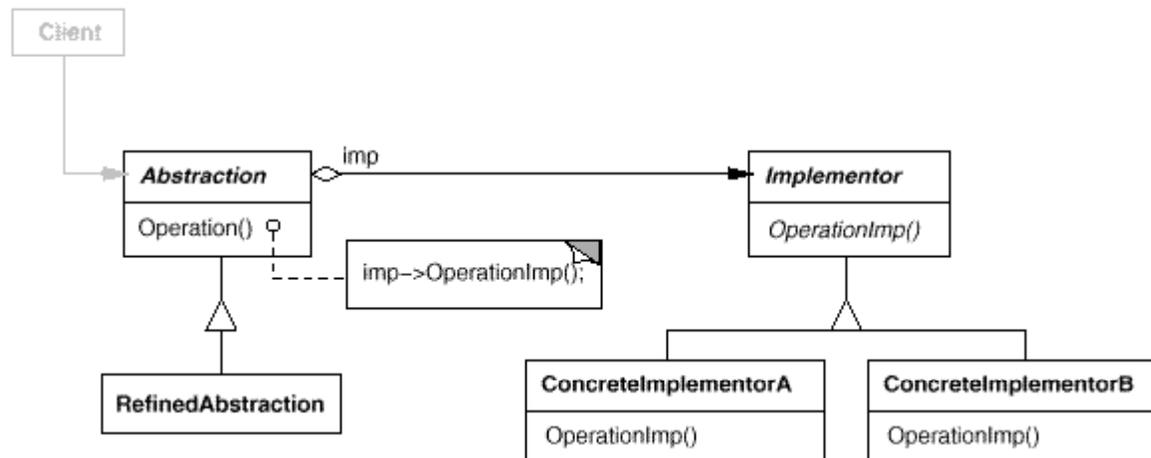
Diagrama del Patrón Descomposición-Sistema



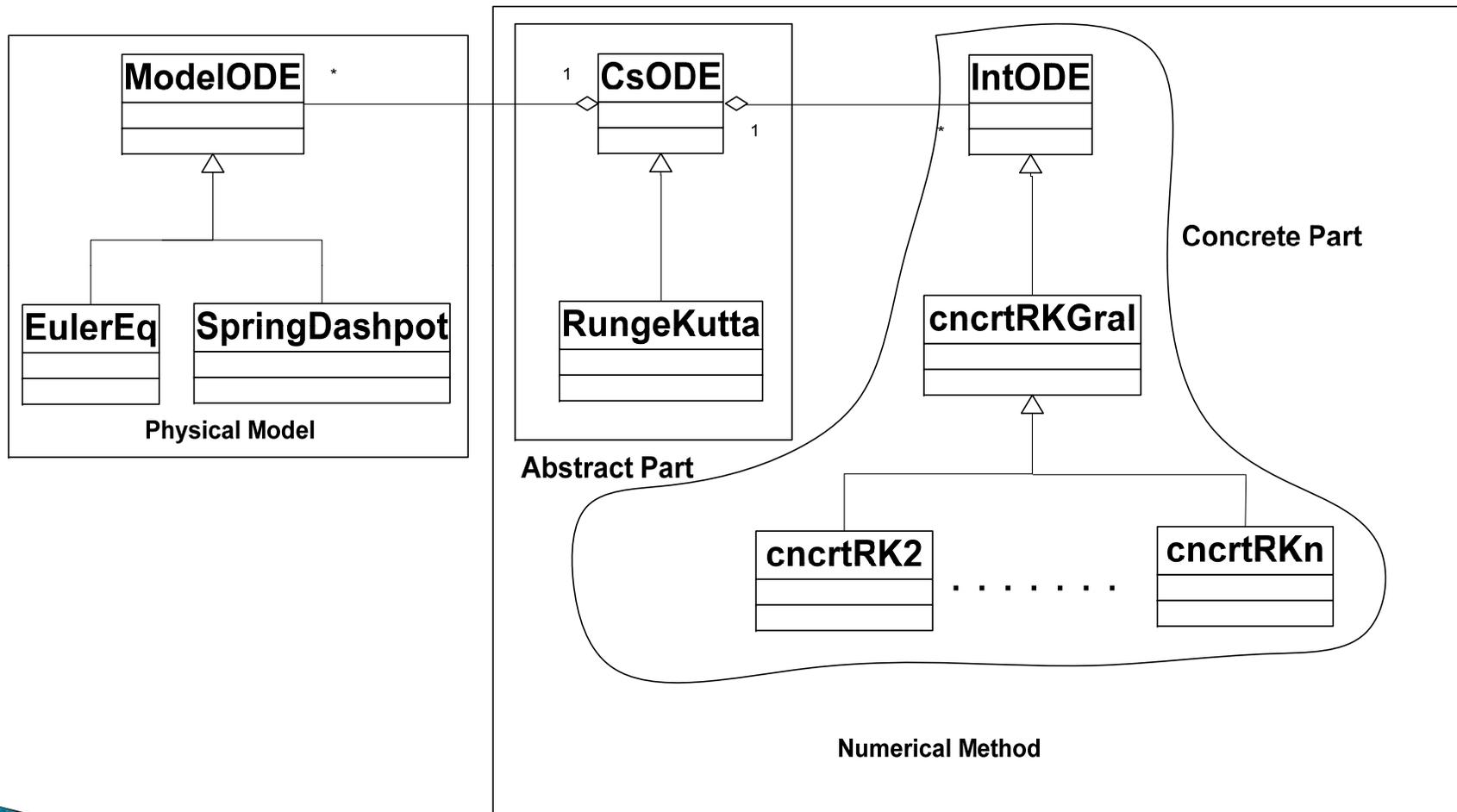
Patrón de Diseño Bridge

► Propósito

- Desacoplar una abstracción de su implementación tal que la dos puedan variar independientemente.



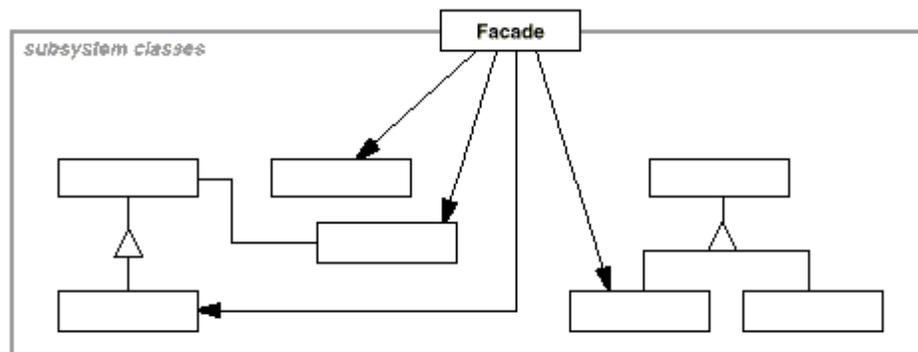
Jerarquía de Clases



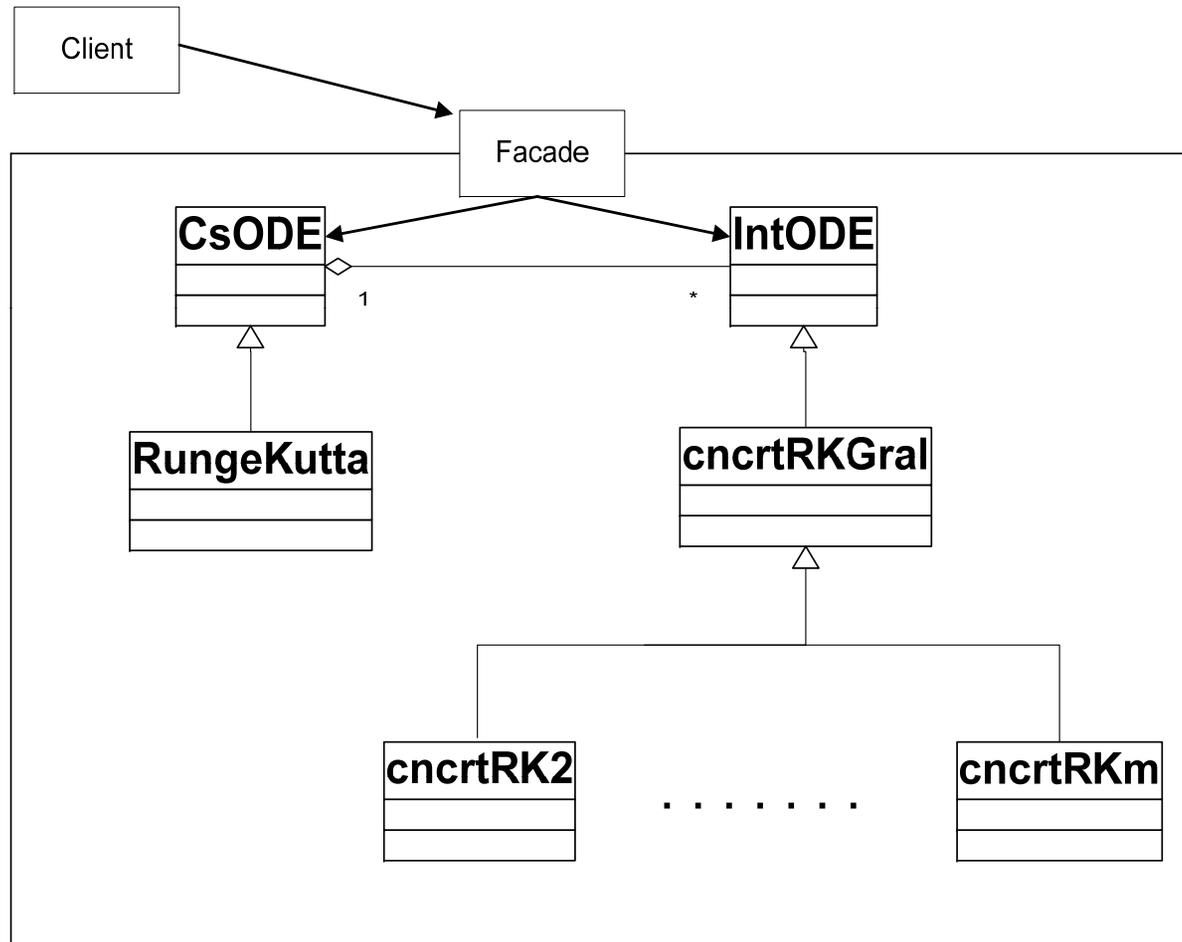
Patrón de Diseño Facade

► Propósito

- Dar una interface unificada a un conjunto de interfaces en un subsistema. Facade define una interface de alto nivel logrando que el uso del subsistema se realice en forma simple.



Ocultamiento del Sistema al Usuario



Conclusiones

- ▶ Se ha propuesto una estrategia para el desarrollo de software científico a través de patrones de cómputo científico.
- ▶ Se establecieron dos nuevos patrones de cómputo científico.
- ▶ La estructura del diseño demuestra que es fácilmente extensible y reusable.
- ▶ En el presente trabajo nos enfocamos más a los atributos de flexibilidad, reusabilidad, escalabilidad, extensibilidad que al de eficiencia.

▶ Gracias

Preguntas