

PEF – 3528 – Ferramentas Computacionais na Mecânica das Estruturas: Criação e Concepção

Prof. Dr. Rodrigo Provasi

e-mail: provasi@usp.br

Sala 09 – LEM – Prédio de Engenharia Civil

Bibliotecas

- Bibliotecas nada mais são que um conjunto de funções (distribuídas em arquivos) que podem ser utilizadas em outros programas.
- No Windows, arquivos que representam esse conjunto de funções têm extensão *dll* (de *Dinamic Link Library*).

Bibliotecas

- São denominadas bibliotecas de vínculo dinâmico por permitir o uso das funções dinamicamente no código.
- Existem diversos tipos de bibliotecas, para os mais variados usos, indo desde bibliotecas de cálculo e formatos de saída até recursos de interface.

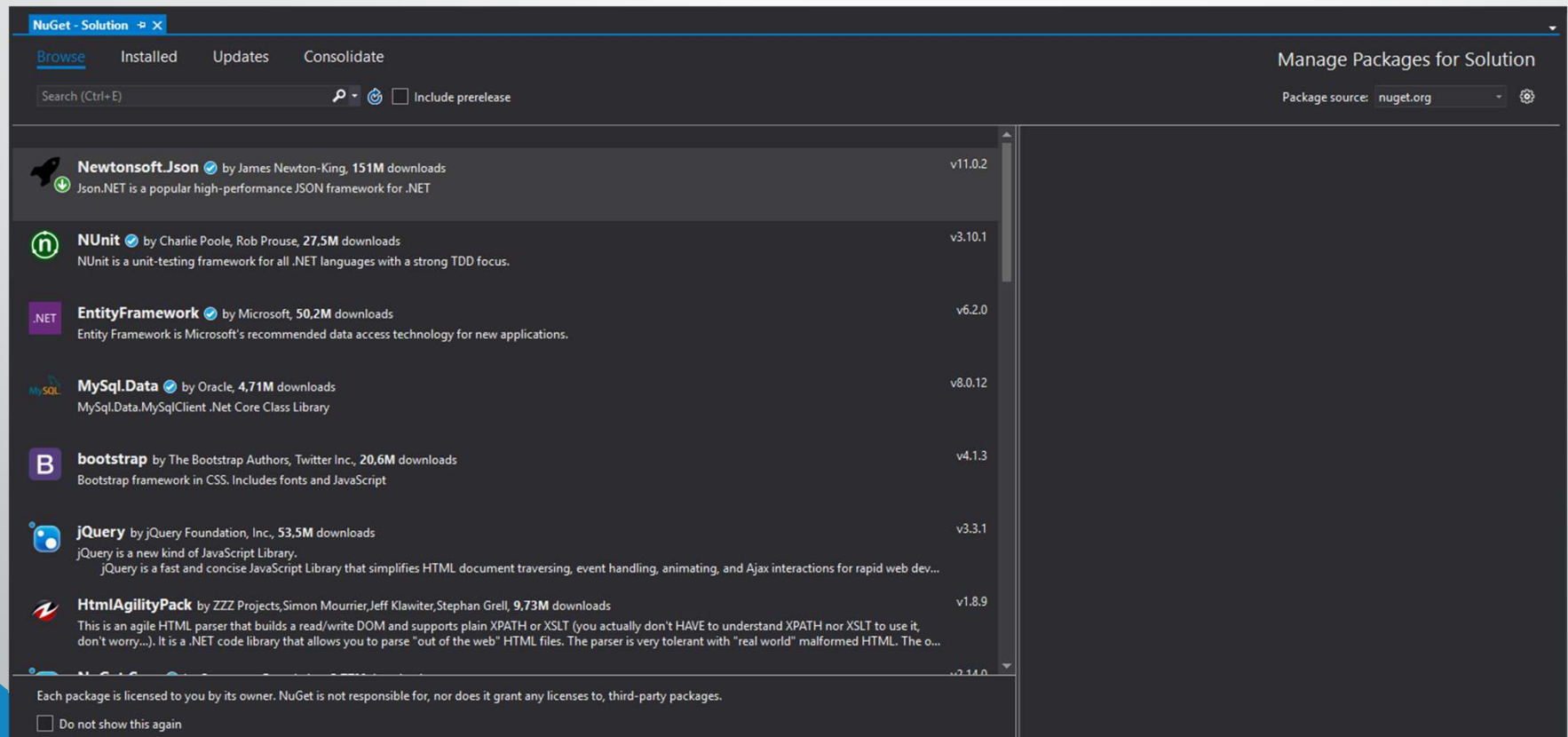
Bibliotecas

- Para usar as bibliotecas, é necessário que seu código indique que a biblioteca será usada.
- No caso do C#, basta adicionar uma referência à biblioteca.
- Ainda, a biblioteca deverá estar no mesmo caminho que o executável.

Bibliotecas no Visual Studio

- O Visual Studio permite que se procure as bibliotecas em um repositório, fazendo a conexão automática com o projeto, bem como colocando os arquivos necessários na pasta de saída do programa.
- Isso se faz através da ferramenta *Nugget*.

Bibliotecas no Visual Studio



The screenshot shows the NuGet Package Manager window in Visual Studio. The window title is "NuGet - Solution". The interface includes tabs for "Browse", "Installed", "Updates", and "Consolidate". The "Browse" tab is active, showing a search bar with the text "Search (Ctrl+E)" and a checkbox for "Include prerelease". The package source is set to "nuget.org".

The list of packages includes:

- Newtonsoft.Json** by James Newton-King, 151M downloads, version v11.0.2. Description: Json.NET is a popular high-performance JSON framework for .NET.
- NUnit** by Charlie Poole, Rob Prouse, 27.5M downloads, version v3.10.1. Description: NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.
- EntityFramework** by Microsoft, 50.2M downloads, version v6.2.0. Description: Entity Framework is Microsoft's recommended data access technology for new applications.
- MySQL.Data** by Oracle, 4.71M downloads, version v8.0.12. Description: MySQL.Data.MySqlClient .Net Core Class Library.
- bootstrap** by The Bootstrap Authors, Twitter Inc., 20.6M downloads, version v4.1.3. Description: Bootstrap framework in CSS. Includes fonts and JavaScript.
- jQuery** by jQuery Foundation, Inc., 53.5M downloads, version v3.3.1. Description: jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web dev...
- HtmlAgilityPack** by ZZZ Projects, Simon Mourrier, Jeff Klawiter, Stephan Grell, 9.73M downloads, version v1.8.9. Description: This is an agile HTML parser that builds a read/write DOM and supports plain XPATH or XSLT (you actually don't HAVE to understand XPATH nor XSLT to use it, don't worry...). It is a .NET code library that allows you to parse "out of the web" HTML files. The parser is very tolerant with "real world" malformed HTML. The o...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again



Bibliotecas

Bibliotecas de Cálculo e Formatos

MathNet Numerics

- Algumas bibliotecas permitem o uso de recursos matemáticos mais avançados dos que podem ser vistos na classe *Math* do C#.
- Como descrito no site dos desenvolvedores:

“Math.NET Numerics aims to provide methods and algorithms for numerical computations in science, engineering and every day use. Covered topics include special functions, linear algebra, probability models, random numbers, interpolation, integration, regression, optimization problems and more.”

MathNet Numerics

- A biblioteca tem uso bem direto. Uma vez adicionada ao projeto (via *NuGet*) basta dizer que se usará o *namespace* adequado e usar as funções normalmente.

MathNet Numerics

```
1: using MathNet.Numerics.LinearAlgebra;
2: using MathNet.Numerics.LinearAlgebra.Double;
3:
4: Matrix<double> A = DenseMatrix.OfArray(new double[,] {
5:     {1,1,1,1},
6:     {1,2,3,4},
7:     {4,3,2,1}});
8: Vector<double>[] nullspace = A.Kernel();
9:
10: // verify: the following should be approximately (0,0,0)
11: (A * (2*nullspace[0] - 3*nullspace[1]))
```

MathNet Numerics

- A biblioteca possui diversos formatos de armazenamento:
 - **Dense Vector** uses a single array of the same length as the vector.
 - **Sparse Vector** uses two arrays which are usually much shorter than the vector. One array stores all values that are not zero, the other stores their indices. They are sorted ascendingly by index.
 - **Dense Matrix** uses a single array in column-major order.
 - **Diagonal Matrix** stores only the diagonal values, in a single array.
 - **Sparse Matrix** stores non-zero values in 3 arrays in the standard compressed sparse row (CSR) format. One array stores all values that are not zero, another array of the same length stores the their corresponding column index. The third array of the length of the number of rows plus one, stores the offsets where each row starts, and the total number of non-zero values in the last field.

MathNet Numerics

- A biblioteca também permite que se resolva sistemas lineares.

```
1: var A = Matrix<double>.Build.DenseFromArray(new double[,] {  
2:     { 3, 2, -1 },  
3:     { 2, -2, 4 },  
4:     { -1, 0.5, -1 }  
5: });  
6: var b = Vector<double>.Build.Dense(new double[] { 1, -2, 0 });  
7: var x = A.Solve(b);
```

MathNet Numerics

- A biblioteca tem suporte à transformadas de Fourier, filtros, ferramentas para obter zeros de funções, entre outros.
- Existem outras bibliotecas de cálculo, tanto gratuitas quanto pagas. Essa serve como sugestão para os alunos.

Newtonsoft.Json

- A biblioteca também conhecida com *Json.net* é uma ferramenta que permite a leitura e gravação de arquivos com extensão *json*, que são no fundo um único dicionário com chaves e valores, cujo valores podem ser outros dicionários ou listas.

NewtonSoft.Json

- A biblioteca permite serialização e deserialização de Json de forma simples e fácil:

```
Product product = new Product();
product.Name = "Apple";
product.Expiry = new DateTime(2008, 12, 28);
product.Sizes = new string[] { "Small" };

string json = JsonConvert.SerializeObject(product);
// {
//   "Name": "Apple",
//   "Expiry": "2008-12-28T00:00:00",
//   "Sizes": [
//     "Small"
//   ]
// }
```

```
string json = @"{
  'Name': 'Bad Boys',
  'ReleaseDate': '1995-4-7T00:00:00',
  'Genres': [
    'Action',
    'Comedy'
  ]
}";

Movie m = JsonConvert.DeserializeObject<Movie>(json);

string name = m.Name;
// Bad Boys
```

NewtonSoft.Json

- Também é possível usar LINQ to Json:

```
JArray array = new JArray();
array.Add("Manual text");
array.Add(new DateTime(2000, 5, 23));

JObject o = new JObject();
o["MyArray"] = array;

string json = o.ToString();
// {
//   "MyArray": [
//     "Manual text",
//     "2000-05-23T00:00:00"
//   ]
// }
```




Bibliotecas

Bibliotecas de Interface

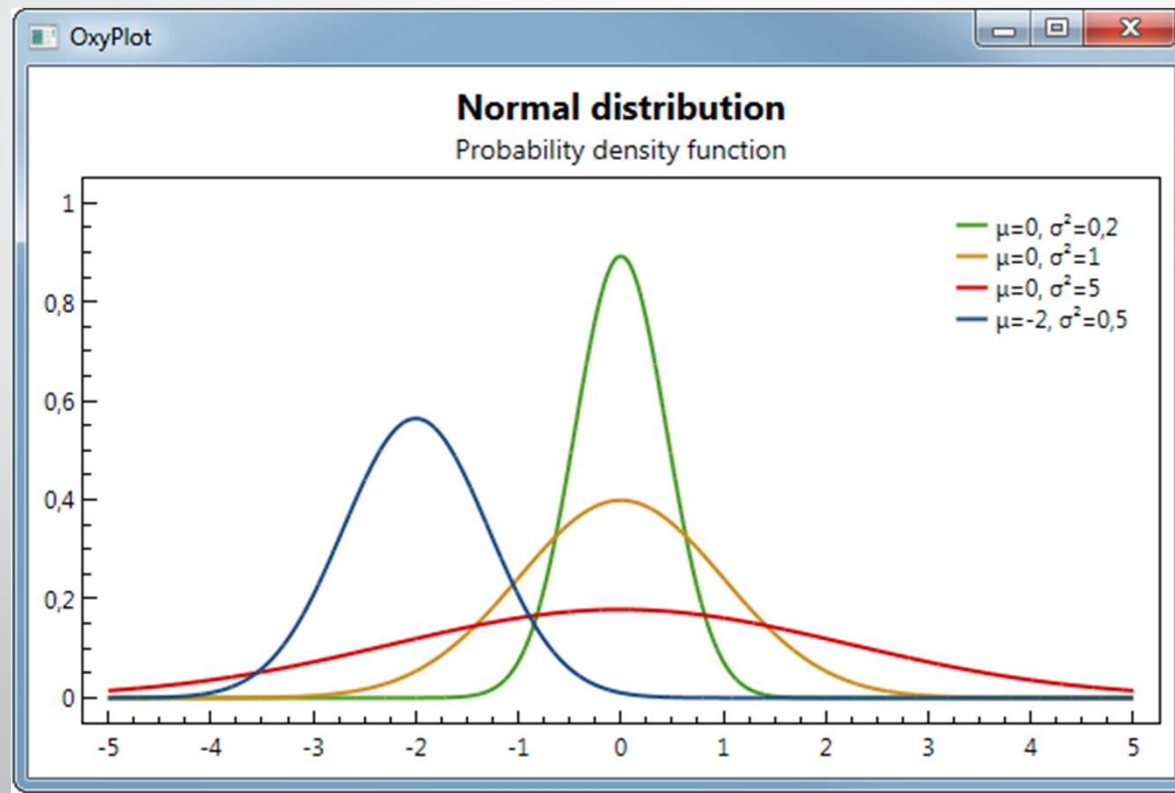
Bibliotecas de Interface

- Apesar de não ter-se discutido ainda os recursos gráficos do C#, algumas bibliotecas serão apresentadas aqui para entender melhor os recursos que podem ser utilizados em um programa.

OxyPlot

- O *OxyPlot* é uma biblioteca que permite a exibição de gráficos.
- Os dados são fornecidos pelo usuário e todas as manipulações (*zoom*, *fit*, *pan* e *probe*) são feitas pelo controle gráfico.

OxyPlot



Avalon Dock

- O *Avalon Dock* é um controle que permite criar regiões para controles serem colocados e permitir que usuário altere durante a execução a posição desses controles.

Avalon Dock

