An easy way to create a website for scientific software project

Gennady Pospelov
Scientific Computing Group at MLZ

Workshop on Neutron Scattering Data Analysis Software
6-8 June, 2018, Sorangna
Choice of technology

Content management systems

or

Static site generators
Content management systems

Generates html dynamically from content written in database

Advantages
- Supports multiple users working in collaborative environment
- Generates html depending on user type
- Allows to add/edit content from the browser

Disadvantages
- Very peculiar workflow
- Crazy learning curve
Content management systems

Generates html dynamically from content written in database

Advantages
- Supports multiple users working in collaborative environment
- Generates html depending on user type
- Allows to add/edit content from the browser

Disadvantages
- Very peculiar workflow
- Crazy learning curve
Static site generators

Generates static html from markdown content

Advantages
- Speed
- Version control for content
- Security
- Less hassle with the hosting

Disadvantages
- No real time content
- No input from bloggers
Welcome to BornAgain

BornAgain is a software package to simulate and fit small-angle scattering at grazing incidence. It supports analysis of both X-ray (GISAXS) and neutron (GISANS) data. Its name, BornAgain, indicates the central role of the distorted wave Born approximation in the physical description of the scattering process. The software provides a generic framework for modeling multilayer samples with smooth or rough interfaces and with various types of embedded nanoparticles.

Read more
Website requirements for scientific software project

Modern
  - Good looking
  - Responsive design
Website requirements for scientific software project

Modern
- Good looking
- Responsive design

Easy to contribute and maintain
- Content focused
- Under git control
- No database
- Familiar workflow
Website requirements for scientific software project

Modern
- Good looking
- Responsive design

Easy to contribute and maintain
- Content focused
- Under git control
- No database
- Familiar workflow

Documentation oriented
- Automatic generation of navigation elements from content
- Code highlight
- Latex support
- Responsive images and image galleries
- Automatic update of site content on new release, versioning
Website requirements for scientific software project

Modern
- Good looking
- Responsive design

Easy to contribute and maintain
- Content focused
- Under git control
- No database
- Familiar workflow

Documentation oriented
- Automatic generation of navigation elements from content
- Code highlight
- Latex support
- Responsive images and image galleries
- Automatic update of site content on new release, versioning

Should still have all attributes of normal website
- Landing, Download, About pages
- Taxonomy
BornAgain

Open-source software package to simulate and fit neutron and x-ray small-angle scattering at grazing incidence.

Its name, BornAgain, indicates the central role of the distorted wave Born approximation in the physical description of the scattering process. The software provides a generic framework for modeling multilayer samples with smooth or rough interfaces and with various types of embedded nanoparticles.

Get started  Download

Currently v1.12.0
## Current release 1.12.0

View changes made in [this release](#).

BornAgain is supported under Windows, Mac OS X and Linux operating systems. For Windows and MacOS we provide binary installer packages, both for Python2 and Python3. For Unix-like operating systems (including Linux and Mac OS X) we support installation from source.

Use links below to download the appropriate package. Previous versions are available [here](#).

<table>
<thead>
<tr>
<th>Operating System</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Windows</strong></td>
<td><strong>MacOS</strong></td>
<td><strong>Linux</strong></td>
</tr>
<tr>
<td>Windows, 64-bit (x86)</td>
<td>Windows, Python3</td>
<td>Windows, Python2</td>
</tr>
<tr>
<td>- <a href="#">Download</a></td>
<td>- <a href="#">Download</a></td>
<td>- <a href="#">Obsolete</a></td>
</tr>
<tr>
<td>Windows 7,8,10 binary installer package, intended for Python3.</td>
<td>Windows 7,8,10 binary installer package, intended for Python2.</td>
<td></td>
</tr>
</tbody>
</table>

To use BornAgain from Python you have to have Python interpreter installed. See [installation instructions](#) for more details.
New release of BornAgain: version 1.12
BornAgain 1.12 has been released

Posted by Walter, on Tue, May 29, 2018
Tags release

BornAgain hotfix: 1.11.1
A bugfix release for BornAgain is now available

Posted by Walter, on Thu, Mar 22, 2018
Tags release

New release of BornAgain: version 1.11
BornAgain 1.11 has been released

Posted by Walter, on Fri, Mar 2, 2018
Tags release

New release of BornAgain: version 1.10
BornAgain 1.10 has been released

Posted by Walter, on Mon, Oct 9, 2017
Tags release

New release of BornAgain: version 1.9
BornAgain 1.9 has been released.

Posted by Walter, on Tue, Jul 4, 2017
Tags release

BornAgain hotfix: 1.8.1
Interference 2D square lattice

Scattering from cylindrical particles distributed along a square lattice.

- Cylinders with radii and heights of 3 nm are deposited on a substrate.
- Because of the presence of the substrate layer the simulation is run using the DWBA.
- The particles are distributed along a square lattice with a lattice length of 25 nm.
- The main axes are parallel to the $x$-axis and $y$-axis of the reference Cartesian frame, respectively.
- The lattice is initialized by placing a cylinder at the origin.
- The incident beam is characterized by a wavelength of 1 Å.
- The incident angles are $\alpha_i = 0.2^\circ$ and $\phi_i = 0^\circ$.

Real-space model

Interference 2D square lattice

```python
import numpy
import bornagain as ba
from bornagain import deg, angstrom, nm

def get_sample():
    """
    Returns a sample with cylinders on a substrate, forming a 2D square lattice
    """
    # defining materials
    m_ambience = ba.HomogeneousMaterial("Air", 0.0, 0.0)
    m_substrate = ba.HomogeneousMaterial("Substrate", 6e-6, 2e-8)
    m_particle = ba.HomogeneousMaterial("Particle", 6e-4, 2e-8)

    # collection of particles
    interference = ba.InterferenceFunction2DLattice.createSquare(25.0*nm)
    pdf = ba.FTDecayFunction2DCauchy(300.0*nm/2.0/numpy.pi, 100.0*nm/2.0/numpy.pi)
    interference.setDecayFunction(pdf)

    cylinder_ff = ba.FormFactorCylinder(3.*nm, 3.*nm)
    cylinder = ba.Particle(m_particle, cylinder_ff)
```
Outline

- Choice of technology
  - Necessary introduction to html and css
  - Markdown + Hugo static site generator
  - BornAgain website and its theme
  - Documentation workflow
Html

Markup language for creating web pages

- Tags to structure the content
  - head, title, body, h1, p, article, nav, figure, img

- Attributes to steer the behavior of certain tag
  
  `<img src="img_girl.jpg" width="500" height="600">`
Html in good old days

- Content is heavily mixed with attributes

```html
<p style="font-size:10px; line-height:13px;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```
Html nowadays

- Content is separated from the presentation

page.html

```html
<p class="highlight">
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```

theme.css

```css
.p.highlight {
    text-align: center;
    font-size: 14px;
    padding: 10px;
    font-style: italic;
    margin: 0;
    display: block;
    border-bottom-right-radius: 5px;
    border-bottom-left-radius: 5px
}
```
CSS frameworks

Collection of css classes for standards-compliant web design

- Grid system for responsive web design
- Web typography
- Unified styling for all elements (lists, tables, buttons, links etc)
- Additional UI elements (accordions, tabs, cards)
Bootstrap css framework

World’s most popular front-end component library

Example of possible layout
Bootstrap css framework

World’s most popular front-end component library

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

<div class="card text-white bg-warning mb-3" style="max-width: 18rem;">
  <div class="card-header">Header</div>
  <div class="card-body">
    <h5 class="card-title">Warning card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
  </div>
</div>
```
Outline

- Choice of technology
- Necessary introduction to html and css
- Markdown + Hugo static site generator
- BornAgain website and its theme
- Documentation workflow
Markdown

- Lightweight markup language with plain text formatting syntax
- Can be converted to HTML and other formats

README.md

```markdown
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
![experimental setup](setup.jpg)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```
Markdown

- Lightweight markup language with plain text formatting syntax
- Can be converted to html and other formats

README.html

```html
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>

<img src="setup.jpg" alt="experimental setup" />

<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```
Markdown

- Lightweight markup language with plain text formatting syntax
- Can be converted to html and other formats

README.html

```
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>

<img src="setup.jpg" alt="experimental setup">

<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</p>
```

+ theme.css to style html can give you a modern looking web page, in theory
Markdown limitations

- Markup language is very poor
- No way to define page layout
Markdown limitations

- Markup language is very poor
- No way to define page layout
- No way to steer individual object appearance
Static site generators

- Generate modern looking web site from markdown content
- Rely on non-markdown inclusions in markdown files

```bash
+++ layout = default +++


{{< figure src="/img/cone6_640.png" class="floatleft">}}

```
Static website generators

**StaticGen**

A List of Static Site Generators for JAMstack Sites

*https://www.staticgen.com*

Contains list of static site generators sorted according to a number of given GitHub stars

**Total number of available generators > 200**

<table>
<thead>
<tr>
<th>Static Site Generator</th>
<th>GitHub Stars</th>
<th>GitHub Issues</th>
<th>GitHub Pull Requests</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jekyll</td>
<td>34431</td>
<td>121</td>
<td>7583</td>
<td>MIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+242</td>
<td>+1</td>
<td>+48</td>
<td>+27</td>
</tr>
<tr>
<td></td>
<td>Deploy to Netlify</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hugo</td>
<td>26047</td>
<td>272</td>
<td>3136</td>
<td>MIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3835</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+592</td>
<td>+16</td>
<td>+42</td>
<td>+193</td>
</tr>
<tr>
<td></td>
<td>Deploy to Netlify</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td>25804</td>
<td>236</td>
<td>2494</td>
<td>MIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+780</td>
<td>+15</td>
<td>+123</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deploy to Netlify</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Languages: Ruby
Templates: Liquid
License: MIT

Languages: Go
Templates: Go
License: Apache 2.0

Languages: JavaScript
Templates: JavaScript
License: MIT
Static website generators

https://www.staticgen.com

Contains list of static site generators sorted according to a number of given GitHub stars

Total number of available generators > 200
Outline

- Choice of technology
- Necessary introduction to html and css
- Markdown + Hugo static site generator
- BornAgain website and its theme
- Documentation workflow
BornAgain website and its theme

```
$ git clone --recurse-submodules https://github.com/scgmlz/BornAgain-website.git
```

- **config.toml**
- **content**
  - about.md
  - contact.md
  - download.md
  - home.md
  - gallery
  - documentation
  - news
- **public**
- **themes**
  - hugo-theme-scgdoc

Website theme

Submodule of https://github.com/scgmlz/hugo-theme-scgdoc

Running command `hugo` in site directory will populate directory `public` with html version of site
# Interference 2D square lattice

Scattering from cylindrical particles distributed along a square lattice.

- Cylinders with radii and heights of 38 nm are deposited on a substrate.
- Because of the presence of the substrate layer the simulation is run using the DWBA.
- The particles are distributed along a square lattice with a lattice length of 258 nm.
- The main axes are parallel to the x-y axis and y-z axis of the reference Cartesian frame, respectively.
- The lattice is initialized by placing a cylinder at the origin.
- The incident beam is characterized by a wavelength of 18.5 nm.
- The incident angles are $\alpha_{\text{inc}} = 0.25$ and $\phi_{\text{inc}} = 0$.

```python
# Interference 2D square lattice

from bornagain import *
import numpy

def get_sample():
    # defining materials
    a_ambient = ba.HomogeneousMaterial('Air', 0.0, 0.0)
    a_substrate = ba.HomogeneousMaterial('Substrate', 6.0, 2.0e8)
    a_particle = ba.HomogeneousMaterial('Particle', 8.0, 2.0e8)

    # collection of particles
    interference = ba.InterferenceFunction2D('InterferenceFunction2D_square', createSquare(258, 258),
        pdf = ba.PDFDecayFunction2DCauchy(300.0, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0)

    interference.setDecayFunction(pdf)
    cylinder = ba.Particle(a_particle, a_substrate, a_ambient)
    cylinder.setCylindricalShape(38.0, 38.0, 38.0)

    return interference, cylinder, print_cylinder, a_ambient

get_sample()
```

[Image: Interference2DSquareLattice.png] [Image: Interference2DSquareLattice_setup.jpg]
hugo-theme-scgdoc

Documentation theme for Hugo developed by our group
https://github.com/scgmlz/hugo-theme-scgdoc

Partials: a single page templates

Custom layout for all markdown files found in ./content/documentation

Shortcodes: markdown extensions
hugo-theme-scgdoc

Documentation theme for Hugo developed by our group
https://github.com/scgmlz/hugo-theme-scgdoc

```
{{ $layout := "col-lg-12" }}
{{- partial "header_page.html" (dict "currentNode" . "layout" $layout) }}

<div class="container">
  <div class="row">
    <div class="col-md-8">
      {{ if .Site.Params.Debug}}theme/layout/documentation/baseof.html{{end}}
      {{ block "main_body" . }}
      {{ end }}
      {{ partial "next-prev-page.html" . }}
    </div>
    <div class="col-md-4">
      {{- partial "sidebar.html" . }}
    </div>
  </div>
</div>

{{- partial "footer.html" . }}
```

*Template file for every *.md found in “documentation”*
Example of Python examples preview

```yaml
+++
title = "Embedded particles"
weight = 10
+++

## Embedded particles

{%% examples-preview %%}
hugo-theme-scgdoc
examples-preview.html shortcode

```html
{{- .Scratch.Set "current" .Page }}

{{- if (.Get 0)}}
  {{- with .Site.GetPage "section" (.Get 0) }}
    {{- $.Scratch.Set "current" . }}
  {{- end }}
{{- end }}

{{- $page := (.Scratch.Get "current") }}

<div class="d-flex flex-column mt-2 mb-4">
  {{- range $page.Pages }}
    <div class="card h-100 my-1">
      <a href="{{$.Permalink}}" class="card-context">
        <div class="card-body">
          <div class="row">
            {{- $resources := (.Resources.ByType "image") }}
            {{- range first 2 $resources }}
              {{- $original := . }}
              {{- $image := ($original.Resize "256x") }}
              <img class="img-fluid" src="{{ $.image.AbsPermalink }}" width="{{ $.image.Width }}" height="{{ $.image.Height }}">
            {{- end }}
          </div>
        </div>
        {{- end }}
        {{- if eq (len $resources) 1 }}
          <div class="col">
            <h6>{{- .Title }} </h6>
          </div>
        {{- end }}
      </div>
    </div>
  {{- end }}
</div>
```

Responsive layout with bootstrap

Automatic generation of thumbnails for faster page load

Loop over all children pages
hugo-theme-scgdoc

Documentation theme for Hugo developed by our group
https://github.com/scgmlz/hugo-theme-scgdoc

Contains

- 900 lines of Hugo templates
- 500 lines of custom css
- 10 lines of custom js

Functionality

- Provides multiple page layout
- Responsive (bootstrap)
- Automatically generates breadcrumb and menu elements from markdown content
- Python highlight (snippets, line numbers, embedded files)
- Responsive galleries
- Automatic generation of page preview
Outline

- Choice of technology
- Necessary introduction to html and css
- Markdown + Hugo static site generator
- BornAgain website and its theme
- Documentation workflow
Documentation workflow

https://github.com/scgmlz/BornAgain-website

master branch: markdown content
gh-pages branch: html of website

GitHub hosting service is configured to show content of gh-pages branch at
https://scgmlz.github.io/BornAgain-website
Documentation workflow

https://github.com/scgmlz/BornAgain-website

`master` branch: markdown content
`gh-pages` branch: html of website

GitHub hosting service is configured to show content of `gh-pages` branch at
https://scgmlz.github.io/BornAgain-website
An easy way to create a website

**Download latest Hugo**
https://github.com/gohugoio/hugo/releases

**Clone our theme**
git clone https://github.com/scgmlz/hugo-theme-scgdoc.git

**Run exampleSite locally, modify it to your taste**

cd hugo-theme-scgdoc/exampleSite
hugo server -D
firefox http://localhost:1313/hugo-theme-scgdoc/