Blog Design Specification

A complete visual identity for a computational physics blog. Set it up once, then focus on writing.

Platform: Quarto

Why Quarto:

- Native support for code blocks, LaTeX math, and figures
- Renders to clean HTML
- JavaScript/Observable support built in
- Works with your existing workflow (Markdown or org-mode export)
- Minimal configuration once set up
- Free GitHub Pages hosting

Emacs workflow: Write in Markdown (or export from org-mode). Quarto handles the rest.

```bash
# Install
# macOS: brew install quarto
# Linux: download from quarto.org

# Create blog
quarto create project blog myblog

# Preview while writing
quarto preview

# Publish to GitHub Pages
quarto publish gh-pages
```

For JavaScript interactives: Quarto supports Observable JS natively. You write JS in code blocks and it just works.

Typography

Font Stack

Body text: Source Serif 4

- Designed for screen reading
- Has the warmth of traditional serifs (Knuth feel)
- Excellent at small sizes
- Free from Google Fonts

Headings: Source Sans 3

- Pairs naturally with Source Serif
- Clean, doesn't compete with content
- Use at heavier weight (600) for hierarchy

Code: JetBrains Mono

- Designed for code readability
- Clear distinction between similar characters (0/O, 1/l/I)
- Free

Math: KaTeX default

- Based on Computer Modern (Knuth's font)
- Quarto uses KaTeX by default
- No configuration needed

Font Sizes

```css
:root {
  --body-font-size: 18px;      /* Comfortable reading */
  --line-height: 1.6;          /* Generous spacing */
  --code-font-size: 15px;      /* Slightly smaller than body */
  --heading-line-height: 1.3;  /* Tighter for headings */
}
```

Hierarchy

- H1: Post title only. 32px.
- H2: Major sections. 24px.
- H3: Subsections. 20px.
- Avoid H4+. If you need them, restructure.

Colors

A restrained palette inspired by Tufte. Muted, high-contrast, easy on the eyes.

Core Palette

```css
:root {
  /* Background and text */
  --bg: #fffff8;              /* Warm off-white (Tufte's choice) */
  --text: #111111;            /* Near-black, not pure black */
  --text-muted: #555555;      /* Secondary text, captions */

  /* Accent - used sparingly */
  --accent: #a00000;          /* Deep red, for links and emphasis */
  --accent-hover: #c00000;    /* Slightly brighter on hover */

  /* Code */
  --code-bg: #f7f7f7;         /* Light gray for code blocks */
  --code-border: #e0e0e0;     /* Subtle border */

  /* Visualization palette (see below) */
}
```

Rules for Using Color

1. **Body text is always --text on --bg.** No exceptions.
2. **Links are --accent.** Underlined. No other decoration.
3. **Code blocks have --code-bg background.** No syntax highlighting colors beyond muted grays and the accent.
4. **Color in visualizations only.** The prose is black and white; diagrams carry the color.

Visualization Style

This is where Red Blob Games and Tufte meet. Clean, informative, no decoration.

Principles

1. **Every pixel serves understanding.** No gradients, shadows, or 3D effects for decoration.
2. **Label directly.** Put text on the graphic, not in a separate legend when possible.
3. **Consistent encoding.** Same color means same thing across all figures in a post.
4. **Interactive when it helps.** Static when it doesn't.

Visualization Color Palette

A set of distinguishable, colorblind-friendly colors for data:

```css
:root {
  --vis-blue: #4269d0;       /* Primary data color */
  --vis-orange: #ff725c;     /* Secondary/contrast */
  --vis-green: #6cc5b0;      /* Tertiary */
  --vis-purple: #9d6bb3;     /* Quaternary */
  --vis-gray: #888888;       /* Neutral/reference */

  /* For sequential data */
  --vis-light: #e8e8e8;      /* Low values */
  --vis-dark: #2a2a2a;       /* High values */

  /* For good/bad, positive/negative */
  --vis-positive: #4a9c6d;   /* Green, muted */
  --vis-negative: #c44e52;   /* Red, muted */
}
```

Static Figures (matplotlib)

Use this matplotlib style for consistency:

```python
```

```
# save as computational_physics.mplstyle

# Figure
figure.figsize: 8, 5
figure.dpi: 150
figure.facecolor: fffff8
figure.edgecolor: fffff8

# Axes
axes.facecolor: fffff8
axes.edgecolor: 111111
axes.linewidth: 0.8
axes.grid: False
axes.spines.top: False
axes.spines.right: False
axes.labelcolor: 111111
axes.labelsize: 12
axes.titlesize: 14

# Ticks
xtick.color: 111111
ytick.color: 111111
xtick.labelsize: 10
ytick.labelsize: 10
xtick.direction: out
ytick.direction: out

# Grid (when used)
grid.color: e0e0e0
grid.linewidth: 0.5
grid.alpha: 0.7

# Lines
lines.linewidth: 2
lines.markersize: 6

# Color cycle
axes.prop_cycle: cycler('color', ['4269d0', 'ff725c', '6cc5b0', '9d6bb3', '888888'])

# Legend
legend.frameon: False
legend.fontsize: 10
```

```
# Font
font.family: sans-serif
font.sans-serif: Source Sans 3, DejaVu Sans, sans-serif
font.size: 11

# Saving
savefig.dpi: 150
savefig.facecolor: fffff8
savefig.edgecolor: fffff8
savefig.bbox: tight
savefig.pad_inches: 0.1
```

Usage:

```python
import matplotlib.pyplot as plt
plt.style.use('computational_physics.mplstyle')
```

Interactive Figures (Observable JS / D3)

For Quarto, use Observable JS blocks:

```javascript
// In your .qmd file:
// ```{ojs}
// Your D3/Observable code here
// ```
```

Style constants to use:

```javascript
const style = {
  bg: "#fffff8",
  text: "#111111",
  accent: "#a00000",
  colors: ["#4269d0", "#ff725c", "#6cc5b0", "#9d6bb3", "#888888"],
  fontFamily: "Source Sans 3, sans-serif",
  fontSize: 14,
  labelFontSize: 12,
};
```

Diagram Guidelines

For explanatory diagrams (the Red Blob Games style):

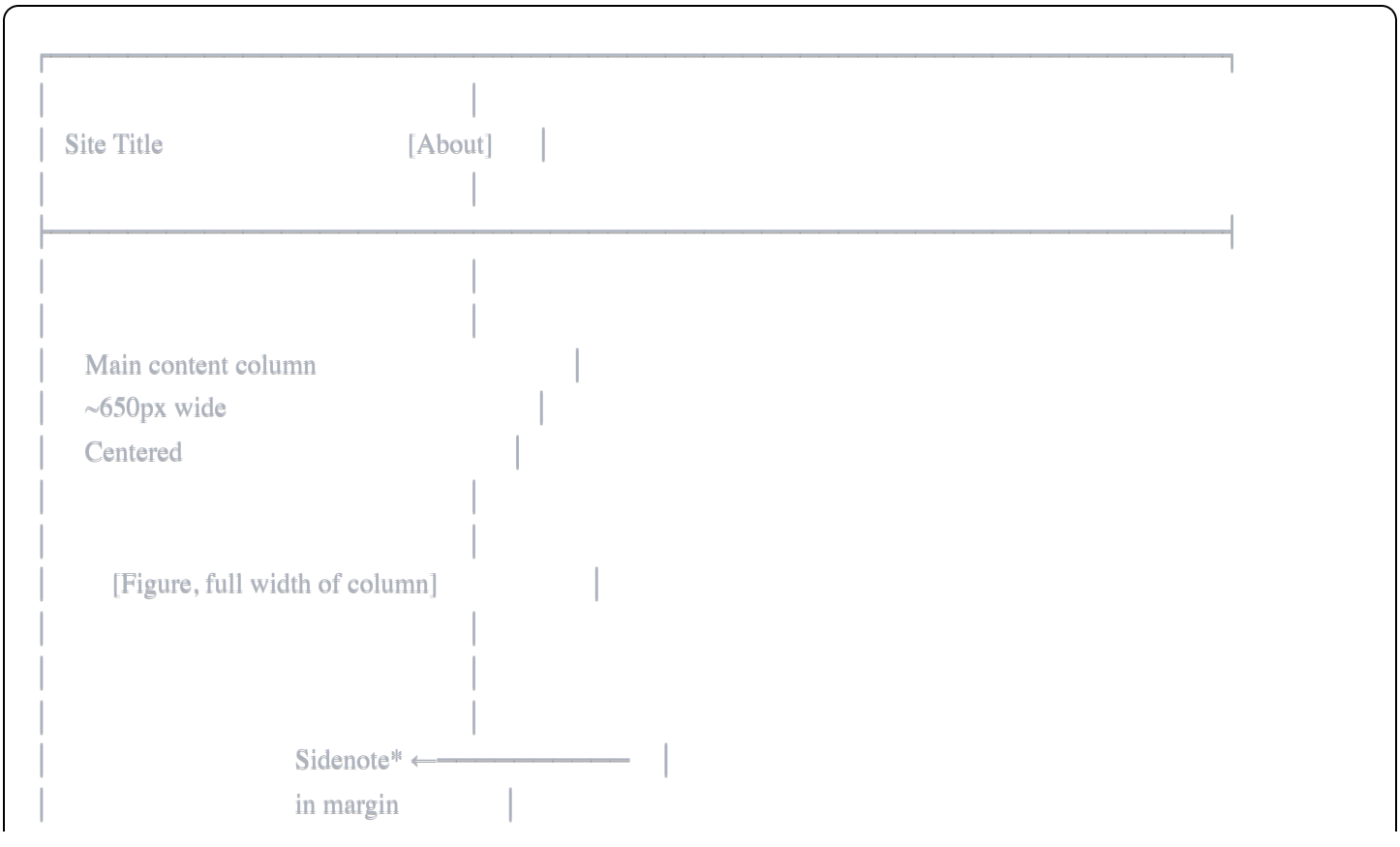
1. **Use SVG.** Vector scales cleanly.
2. **Thick lines.** 2-3px strokes minimum.
3. **Large labels.** 14px+ for readability.
4. **Generous whitespace.** Don't cram.
5. **Arrows with purpose.** Show direction, causation, flow.
6. **Grid/axes only when necessary.** Remove if not referenced.

Tools:

- **Excalidraw** for quick hand-drawn style diagrams
- **D3.js** for programmatic/interactive diagrams
- **Inkscape** if you need precise vector editing
- **matplotlib** for plots and charts

Layout

Page Structure



*Sidenotes are a Tufte signature. Quarto supports them.

Content Width

- **Body text:** 650px max (optimal line length for reading)
- **Figures:** Can extend wider (up to 900px) when needed
- **Sidenotes:** In the right margin on wide screens, inline on mobile

Navigation

Minimal. Two links maximum in header:

- Site title (links home)
- "About" page

Posts listed on home page, reverse chronological. No categories, tags, or archives unless you accumulate enough posts to need them (you won't for a while).

Post Structure

markdown

```
---
title: "Post Title"
date: 2025-02-15
description: "One sentence summary"
---

Opening paragraph. No heading needed.

## First Major Section

Content...

## Second Major Section

Content...

## Conclusion (or nothing)

Maybe a brief wrap-up. Or just end when you're done.
```

No "Introduction" heading. Just start.

Quarto Configuration

_quarto.yml

```yaml
```

```yaml
project:
  type: website
  output-dir: docs

website:
  title: "Your Name"
  navbar:
    left:
      - href: index.qmd
        text: Home
      - href: about.qmd
        text: About
  page-footer:
    center: "© 2025 Your Name"

format:
  html:
    theme:
      light: custom.scss
    css: styles.css
    toc: true
    toc-depth: 2
    toc-location: right
    code-fold: false
    code-tools: false
    highlight-style: github
    fontsize: 18px
    linestretch: 1.6
    mainfont: "Source Serif 4"
    monofont: "JetBrains Mono"
```

custom.scss

```scss
```

```scss
/*-- scss:defaults --*/

$body-bg: #fffff8;
$body-color: #111111;
$link-color: #a00000;
$font-family-sans-serif: "Source Sans 3", sans-serif;
$font-family-serif: "Source Serif 4", serif;
$font-family-monospace: "JetBrains Mono", monospace;

/*-- scss:rules --*/

body {
  font-family: $font-family-serif;
  background-color: $body-bg;
  color: $body-color;
}

h1, h2, h3, h4, h5, h6 {
  font-family: $font-family-sans-serif;
  font-weight: 600;
}

a {
  color: $link-color;
  text-decoration: underline;
  text-underline-offset: 2px;
}

a:hover {
  color: darken($link-color, 10%);
}

code {
  font-family: $font-family-monospace;
  font-size: 0.85em;
  background-color: #f7f7f7;
  padding: 0.1em 0.3em;
  border-radius: 3px;
}

pre {
  background-color: #f7f7f7;
  border: 1px solid #e0e0e0;
```

```
  border-radius: 4px;
  padding: 1em;
}

.quarto-figure {
  margin: 2em 0;
}

figcaption {
  font-size: 0.9em;
  color: #555555;
  margin-top: 0.5em;
}

/* Sidenotes / margin notes */
.column-margin {
  font-size: 0.85em;
  color: #555555;
}
```

styles.css

```
css
```

```css
/* Additional custom styles */

/* Constrain content width */
.content {
  max-width: 650px;
}

/* Allow figures to be wider */
.quarto-figure-center {
  max-width: 900px;
  margin-left: auto;
  margin-right: auto;
}

/* Remove number from title */
.header-section-number {
  display: none;
}

/* Clean up code blocks */
.sourceCode {
  font-size: 15px;
}

/* Observable/JS output styling */
.observablehq {
  font-family: "Source Sans 3", sans-serif;
}
```

Workflow Summary

Writing a Post

1. Create `posts/my-post/index.qmd`
2. Write in Markdown with LaTeX math (`$...$`, `$$...$$`)
3. Add code blocks that execute (Python, Julia, JS)
4. Add figures (matplotlib saves, or inline Observable)
5. Preview: `quarto preview`
6. When ready: `quarto publish gh-pages`

Adding an Interactive Visualization

markdown

````
## Interactive Demo

Here's the system in action:

```{ojs}
// Observable JS code here
viewof parameter = Inputs.range([0, 10], {value: 5, step: 0.1, label: "Parameter"})

{
  const svg = d3.create("svg")
    .attr("viewBox", [0, 0, 600, 400])
    .style("background", "#fffff8");

  // Your visualization code...

  return svg.node();
}
```
````

Adding a Static Figure

python

```
#| label: fig-lorenz
#| fig-cap: "The Lorenz attractor"

import matplotlib.pyplot as plt
plt.style.use('computational_physics.mplstyle')

# Your plotting code...

plt.show()
```

What This Gives You

- **Consistent look** without thinking about it
- **Knuth feel** via serif body text and Computer Modern math

- **Tufte influence** via restrained color, sidenotes, high data-ink ratio
- **Red Blob Games capability** via Observable JS for interactives
- **Minimal maintenance** — Quarto handles the machinery
- **Focus on writing** — the tools get out of your way

Files to Create

1. `_quarto.yml` — site configuration
2. `custom.scss` — theme overrides
3. `styles.css` — additional styling
4. `computational_physics.mplstyle` — matplotlib style
5. `index.qmd` — home page (list of posts)
6. `about.qmd` — about page
7. `posts/` directory — your posts go here

Set this up once. Then just write.