MagVis Software Mathematical Foundation

https://github.com/najikrayem/magvis

1 Terms

- Use polyline representation of wire centerline path $\{l_0, l_1, l_2, ..., l_n\}$, or convert parametric curve into polyline.
- Let P be the magnetic field evaluation point.
- Let I be current in Amps.
- Let a be wire radius (used for singularity issue).

For each segment edge $[l_u, l_{u+1}]$ of n segments total:

- Let δl be the segment vector $\delta l = l_{u+1} - l_u$.
- Let m be the quadrature point (mid point).
- Let R be the evaluation point vector $R = P - m$.
- Let $r = ||R||$.
- Let B be magnetic field vector at P.

2 Underlying Main Algorithm

Algorithm 1 Underlying Main Algorithm

$B \leftarrow (0, 0, 0)$
for $i = 0$ to n **do**
    $numerator \leftarrow I(\delta l \times R)$
    $denominator \leftarrow (r^2 + a^2)^{3/2}$
    ▷ a term prevents explosion if P is close to wire
    $B \leftarrow B + \frac{numerator}{denominator}$
end for
    return $B \times 1e - 7$

3 Level-Of-Detail (LOD) Optimization Algorithm

The optimization has two components, greedy component and LOD component:

3.1 Greedy

If a section of the wire satisfies these criteria:

- Is made up of several segments in a completely straight path, OR
- The angle between any two consecutive segments does not exceed 0.05, AND
- The total radius of the curve does not exceed 1/8 of r, AND
- The distance between the first and last points in the curve is less than r, then

the section is considered straight. If P falls on the line plane, then only a single efficient closed-form calculation is necessary for straight sections:

$$B = \frac{\mu_0 I}{2\pi P}(\sin\theta_0 + \sin\theta_1)$$

where θ_0 and θ_1 are angles with respect to the ends of the segment.

3.2 LOD

For non-straight curves, the main algorithm is used, with a LOD optimization. This optimization relies on a preset absolute error bound threshold ϵ. The optimization is pretty simple:

1. Do the per-segment calculation for one segment and store the result in accumulator B_a.
2. Double the detail of the segment by splitting it into two segments that reduces error against the original curve.
3. Do the per-segment calculation for the two new segments and combine the results in accumulator B_b.
4. Repeat recursively until the difference between B_a and B_b converges to ϵ.

In the future, a heuristic can be employed to enhance the initial LOD level and recursion step size.