Experimenting Iterative Methods for Inverse Problems at Low Precision Levels

Riley Chen, Kristina Gong, Zoe Ji
Emory Summer REU 2022
Emory University
Atlanta, GA, USA
Outline

1. Chop
2. CGLS
3. Experiment
Chop: Overview

A closer look at double, single, fp16 precision:

Format of Floating points
IEEE754

64bit = double, double precision
1 11bit  52bit

32bit = float, single precision
1 8bit  23bit

16bit = half, half precision
1 5bit  10bit

---

Chop: Overview

- Simulate low-precision arithmetics
- Need to chop each operation

```javascript
options.format = 'fp16';
chop([],options);

x = chop(x);
y = chop(y);
z = chop(z);
s = chop(x + chop(y * z));

options.format = 'c';
options.params = [11,23]
chop([],options);
```
Chop: Blocking

- Break an inner product into several smaller inner products
- Compute them independently and then sum

\[
\begin{align*}
x & \equiv [x_1, x_2, x_3, x_4, x_5, x_6] \\
y & \equiv [y_1, y_2, y_3, y_4, y_5, y_6] \\
x_{-1} & \equiv [x_1, x_2, x_3] \\
y_{-1} & \equiv [y_1, y_2, y_3] \\
x_{-2} & \equiv [x_4, x_5, x_6] \\
y_{-2} & \equiv [y_4, y_5, y_6]
\end{align*}
\]
Chop: Blocking

**error vs block size**

![Plot showing average relative error vs block size for different precision levels.](image)

**error vs block size at fp16**

![Plot showing average relative error vs block size at fp16 for different block sizes.](image)
Matrix-vector multiplication:

Instead of:
\[ a = \text{chop}(a + \text{chop}(\text{chop}(X(:,j))*\text{chop}(y(j)))) \]

```matlab
function C = mv_blocked(X,y,block_size)
    % compute the product of a matrix and a vector with chop
    A = chop(chop(X).*chop(y'));
    if nargin < 3
        block_size = 256; % default block size
    end
    [m, n] = size(X);
    k = floor(n/block_size);
    C = zeros(m,1);
    for i = 1:k
        a=zeros(m,1);
        for j = (i-1)*block_size+1 : i*block_size
            a = chop(a+A(:,j));
        end
        C = chop(C + a);
    end
    if n-k*block_size ~= 0
        b=zeros(m,1);
        for i = k*block_size+1:n
            b = chop(b+ A(:,i));
        end
        C = chop(C + b);
    end
```
CGLS: Overview

- Conjugate Gradient Method: Solve $Ax = b$ for SPD matrices
- CGLS:
  - Generalize to all the matrices
  - $A \rightarrow A^T A$, $b \rightarrow A^T b$ without explicitly calculating $A^T A$
Algorithm 7.4.1. CGLS. Let $x^{(0)}$ be an initial approximation, set
\[ r^{(0)} = b - Ax^{(0)}, \quad p^{(0)} = s^{(0)} = A^T r^{(0)}, \quad \gamma_0 = \| s^{(0)} \|^2, \]
and for $k = 0, 1, 2, \ldots$ while $\gamma_k > \text{tol}$ compute
\[
q^{(k)} = Ap^{(k)}, \\
alpha_k = \gamma_k / \| q^{(k)} \|^2, \\
x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \\
r^{(k+1)} = r^{(k)} - \alpha_k q^{(k)}, \\
s^{(k+1)} = A^T r^{(k+1)}, \\
\gamma_{k+1} = \| s^{(k+1)} \|^2, \\
beta_k = \gamma_{k+1} / \gamma_k, \\
p^{(k+1)} = s^{(k+1)} + \beta_k p^{(k)}.
\]


Chop each operation! :)

\[
Ax = \text{mv\_blocked}(A, x); \quad \text{normr2} = \text{vv\_blocked}(d(:,),d( : )); \\
d = \text{mv\_blocked}(A’, b); \\
d = \text{chop}(d - \text{mv\_blocked}(A’, Ax)); \\
\text{normr2} = \text{vv\_blocked}(d(:,),d( : )); \\
Ad = \text{mv\_blocked}(A, d); \\
\text{alpha} = \text{chop}(\text{normr2}/\text{normAd2}); \\
x = \text{chop}(x + \text{chop}(\text{alpha*d})); \\
r = \text{chop}(r - \text{chop}(\text{alpha*Ad})); \\
s = \text{mv\_blocked}(A’, r); \\
q = s; \quad \text{normr2\_new} = \text{vv\_blocked}(q,q); \\
\text{beta} = \text{chop}(\text{normr2\_new}/\text{normr2}); \\
d = \text{chop}(s + \text{chop}(\text{beta*d}));
\]
Experiment: Image Deblurring (no noise)

Figure: Double precision problem size 64 with mild blurring.

Figure: Single precision problem size 64 with mild blurring.

Figure: Half (fp16) precision problem size 64 with mild blurring.
Experiment: Image Deblurring (no noise)

- The error norm:

Figure: The error norm of a size 64 problem with mild blurring of different precisions.
Experiment: Image Deblurring (with noise)

-Single

Figure: Single precision, problem size 64 with mild blurring and 0.1% noise.

Figure: Single precision, problem size 64 with mild blurring and 1% noise.

Figure: Single precision, problem size 64 with mild blurring and 10% noise.
Experiment: Image Deblurring (with noise)

- Half

**Figure:** Half precision, problem size 64 with mild blurring and 0.1% noise.

**Figure:** Half precision, problem size 64 with mild blurring and 1% noise.

**Figure:** Half precision, problem size 64 with mild blurring and 10% noise.
Experiment: Image Deblurring (with noise)

- Error norm

**Figure:** Error norm for problem size 64 with mild blurring and 0.1% noise.

**Figure:** Error norm for problem size 64 with mild blurring and 1% noise.

**Figure:** Error norm for problem size 64 with mild blurring and 10% noise.

Riley Chen, Kristina Gong, Zoe Ji
Experiment: Tomography (no noise)

Figure: Double precision problem size 64 with default blurring.

Figure: Single precision problem size 64 with default blurring.

Figure: Half (fp16) precision problem size 64 with default blurring.

Riley Chen, Kristina Gong, Zoe Ji
Experiment: Tomography (no noise)

- Why?

- NaNs!
Algorithm 7.4.1. CGLS. Let \( x^{(0)} \) be an initial approximation, set
\[
    r^{(0)} = b - Ax^{(0)}, \quad p^{(0)} = s^{(0)} = A^T r^{(0)}, \quad \gamma_0 = \|s^{(0)}\|_2^2,
\]
and for \( k = 0, 1, 2, \ldots \) while \( \gamma_k > \text{tol} \) compute
\[
    q^{(k)} = Ap^{(k)}, \\
    \alpha_k = \gamma_k / \|q^{(k)}\|_2^2, \\
    x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \\
    r^{(k+1)} = r^{(k)} - \alpha_k q^{(k)}, \\
    s^{(k+1)} = A^T r^{(k+1)}, \\
    \gamma_{k+1} = \|s^{(k+1)}\|_2^2, \\
    \beta_k = \gamma_{k+1} / \gamma_k, \\
    p^{(k+1)} = s^{(k+1)} + \beta_k p^{(k)}.
\]

\( \gamma \) becomes Inf, there is an overflow.

\(^3\text{Björck, Numerical methods for least squares problems}\)
Experiment: Tomography (no noise)

Our solution: $A \rightarrow A/100$, $b \rightarrow b/100$
Experiment: Tomography

- Error norm:

**Figure:** Error norm of size 64 problem with 0 noise at different precision level.

**Figure:** Error norm of size 64 problem with 1% noise at different precision level.

**Figure:** Error norm of size 64 problem with 10% noise at different precision level.
Experiment: Tomography (with noise)

**Figure:** Single precision problem size 64 with zero noise.

**Figure:** Single precision problem size 64 with 1% noise.

**Figure:** Single precision problem size 64 with 10% noise.
γ becomes Inf in the original problem, the overflow results in NaNs from the first iteration

After we divide both A and b by 10, \( \|q\|_2 = \text{Inf} \), \( \alpha = x = 0 \) in the first iteration. Later no underflow or overflow occurs, yet plot is always blur

**Figure:** fp16 problem size 32 with zero noise
Experiment: Tomography (some interesting cases)

\begin{algorithm}
\textbf{Algorithm 7.4.1. CGLS.} Let $x^{(0)}$ be an initial approximation, set
\[ r^{(0)} = b - Ax^{(0)}, \quad p^{(0)} = s^{(0)} = A^T r^{(0)}, \quad \gamma_0 = \|s^{(0)}\|_2, \]
and for $k = 0, 1, 2, \ldots$ while $\gamma_k > \text{tol}$ compute
\[
\begin{align*}
q^{(k)} &= Ap^{(k)}, \\
\alpha_k &= \gamma_k/\|q^{(k)}\|_2, \\
x^{(k+1)} &= x^{(k)} + \alpha_k p^{(k)}, \\
r^{(k+1)} &= r^{(k)} - \alpha_k q^{(k)}, \\
s^{(k+1)} &= A^T r^{(k+1)}, \\
\gamma_{k+1} &= \|s^{(k+1)}\|_2, \\
\beta_k &= \gamma_{k+1}/\gamma_k, \\
p^{(k+1)} &= s^{(k+1)} + \beta_k p^{(k)}.
\end{align*}
\end{algorithm}

$\|q\|_2^2 = \infty$, $\alpha = x = 0$ in the first iteration.

\[4\] Björck, \textit{Numerical methods for least squares problems}
Experiment: Tomography (some interesting cases)

- $\gamma$ becomes $\text{Inf}$ in the original problem, the overflow results in NaNs from the first iteration.

- We set $A \rightarrow A/100$, and $b \rightarrow b/100$. This is the last iteration with all Inf and -Infs before NaN occurs.

**Figure:** fp16 problem size 32 with default blur and zero noise, 14th iteration

**Figure:** fp16 problem size 32 with default blur and zero noise, 13th iteration

Riley Chen, Kristina Gong, Zoe Ji
Simulating low precision
Where We Will Go Next...

- Run experiments of larger sizes
- Implement other iterative methods that avoid inner products to eliminate NaNs
Bibliography