

## Re: Code vectorization

---

*Source:* <http://newsgroups.derkeiler.com/Archive/Comp/comp.soft-sys.matlab/2008-12/msg01578.html>

---

- *From:* "Steven Lord" <[slord@xxxxxxxxxxxxx](mailto:slord@xxxxxxxxxxxxx)>
  - *Date:* Mon, 8 Dec 2008 10:33:07 -0500
- 

"Pete" <[petematlab@xxxxxxxxx](mailto:petematlab@xxxxxxxxx)> wrote in message  
[news:ghgbh3\\$N8v\\$1@xxxxxxxxxxxxxxxxxxxxxxxx](news:ghgbh3$N8v$1@xxxxxxxxxxxxxxxxxxxxxxxx)

Hi all

Although I have been using matlab on and off for a fair while, for various little things, I today had to try some numerical integration for the first time. I wanted to integrate a normal distribution between various limits. I first tried a test using 'normspec':

```
p_1 = normspec([-1.8 1.8], 0, 1)
p_1 =
0.9281
```

Next, I tried the same calculation using 'quadl' (and quad):

```
gauss = @(z) 1/sqrt(2*pi)*exp(-z.^2/2);
p_2 = quadl(gauss,-1.8,1.8)
p_2 =
0.9281
```

Same answer, so I felt like I was on the right lines. But now I would like to integrate using vectors of lower and upper limits, e.g.

```
lower = [-1 -2 -3];
upper = abs(lower);
p_3 = quadl(gauss,lower,upper) which of course doesn't work
p_4 = quadv(gauss,lower,upper) nope!
```

I also tried symbolic integration and the 'eval' function, but again could not get it to accept vectors of limits. I also used 'normcdf' but I wish later to perform double integrations, i.e. calculate layers in a normal distribution, and I was not sure how to do that with that command, so I have stuck with quad. I have little doubt I am doing something fundamentally wrong here. Any pointers/solutions in how to vectorize such a calculation would be a massive help. I have had a fair dig around in the 'help' section and on the net, but think I am lacking some basic knowledge on how to use these functions.

## Re: Code vectorization

No, you're not doing anything wrong here. None of the quadrature routines support vectors as the limits. Keep in mind that the quadrature routines work by breaking up the region over which you're integrating into smaller subregions and approximate the integrals on those subregions, breaking up the subregions if the approximation is not accurate enough. If the quadrature routines did support vectors of limits, what would happen if the integral was accurately approximated on a subinterval for one set of limits but not other sets?

- \* Breaking apart the subregion for all the sets of limits would mean we were doing extra calculation on the set that had already succeeded.
- \* Not breaking about the subregion for any of the sets of limits would mean we would fail to perform the integration for the sets of limits that had not already succeeded.
- \* Breaking apart the region for some but not all sets of limits would require a lot of bookkeeping that would probably slow down the quadrature functions.

Since you've already said that your version of MATLAB doesn't support ARRAYFUN, a FOR loop is probably your best option. Don't forget to preallocate!

```
lower = [-1 -2 -3];
upper = abs(lower);

% preallocation
p_3 = zeros(size(lower));
for k = 1:numel(lower)
    p_3(k) = quadl(gauss, lower(k), upper(k));
end
```

--  
Steve Lord  
slord@xxxxxxxxxxxxxx

.