Add element to vector

Having a vector `x` and I have to add an element (`newElem`) .

Is there any difference between -

```
x(end+1) = newElem;
```

and

```
x = [x newElem];
```

?

matlab

edited Jan 21 at 10:53
Dan
21.8k 4 19 50

asked Apr 24 '13 at 9:13
URL87
1,967 11 33 77

3 Answers

`x(end+1) = newElem` is a bit more robust.

`x = [x newElem]` will only work if `x` is a row-vector, if it is a column vector `x = [x; newElem]` should be used. `x(end+1) = newElem` , however, works for both row- and column-vectors.

In general though, growing vectors should be avoided. If you do this a lot, it might bring your code down to a crawl. Think about it: growing an array involves allocating new space, copying everything over, adding the new element, and cleaning up the old mess...Quite a waste of time if you knew the correct size beforehand :)

edited Apr 24 '13 at 9:38
Rody Oldenhuis
25.5k 4 19 59

answered Apr 24 '13 at 9:15
ThijsW
1,674 5 16

2 Also for the second method, `x` must be initialized first! – Dan Apr 24 '13 at 9:22

1 @RodyOldenhuis, no problem! I did the same, probably at the same time. @Dan, that's true, but as the question mentioned "having a vector `x` (of size `n`)", I kind of assumed `n` to be non-zero and the vector being initialized already :) – ThijsW Apr 24 '13 at 9:24

Yes that's true – Dan Apr 24 '13 at 9:25

1 @ThijsW: Still, using `end` rather than some variable `n` (which might be a `global` for all you know!) is the more universal, robust-no-cost way to go – Rody Oldenhuis Apr 24 '13 at 9:37

Just to add to @ThijsW's answer, there is a significant speed advantage to the first method over the concatenation method:

```
big = 1e5;
tic;
x = rand(big,1);
toc

x = zeros(big,1);
tic;
for ii = 1:big
    x(ii) = rand;
end
toc

x = [];
tic;
for ii = 1:big
    x(end+1) = rand;
end;
toc

x = [];
tic;
for ii = 1:big
    x = [x rand];
```

```
end;
toc
```

```
   Elapsed time is 0.004611 seconds.
   Elapsed time is 0.016448 seconds.
   Elapsed time is 0.034107 seconds.
   Elapsed time is 12.341434 seconds.
```

I got these times running in 2012b however when I ran the same code on the same computer in matlab 2010a I get

```
Elapsed time is 0.003044 seconds.
Elapsed time is 0.009947 seconds.
Elapsed time is 12.013875 seconds.
Elapsed time is 12.165593 seconds.
```

So I guess the speed advantage only applies to more recent versions of Matlab

edited Apr 24 '13 at 9:47

answered Apr 24 '13 at 9:20

Dan
21.8k 4 19 50

+1, Edited to add the obvious as well. I'll test again on a "real" CPU (I'm on this crappy unreliable no-good APU thing now...) – Rody Oldenhuis Apr 24 '13 at 9:28

There, all better now :) – Rody Oldenhuis Apr 24 '13 at 9:30

@Dan, same for me, I get 0.028 for the 3rd option and 8.909 for the last – ThijsW Apr 24 '13 at 9:31

1 I also think the JIT optimisation for the `x(end+1)` case is a pretty recent addition (R2012a or so...). I vagualey remember reading something like that in some changenotes at smoe point. I also get very different results on my APU/Matlab R2010, but I'm not sure if that's due to the Matlab version or the APU... – Rody Oldenhuis Apr 24 '13 at 9:33

@RodyOldenhuis and ThiijsW see my recent comparison between older and newer matlab – Dan Apr 24 '13 at 9:33

As mentioned before, the use of `x(end+1) = newElem` has the advantage that it allows you to concatenate your vector with a scalar, regardless of whether your vector is transposed or not. Therefore it is more robust for adding scalars.

However, what should not be forgotten is that `x = [x newElem]` will also work when you try to add multiple elements at once. Furthermore, this generalizes a bit more naturally to the case where you want to concatenate matrices. `M = [M M1 M2 M3]`

All in all, if you want a solution that allows you to concatenate your existing vector `x` with `newElem` that may or may not be a scalar, this should do the trick:

```
x(end+(1:numel(newElem)))=newElem
```

edited Jan 22 '14 at 15:03

answered Apr 24 '13 at 19:27

Dennis Jaheruddin
12.2k 2 18 52

1 I think your last example should be: x(end+1:end+length(newElem)) = newElem – Digna Jan 22 '14 at 15:00

@Digna Thanks for finding the bug, I have updated the answer to fix the problem. – Dennis Jaheruddin Jan 22 '14 at 15:07

With my Matlab2011b, there was also a drastic (~50x) speed improvement on vector concatenation with this method vs. the a=[a b] method. – JaBe Apr 8 at 17:17